

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**ROGERIO MALTA BRANCO**

**AGENDAMENTO DE TAREFAS EM SISTEMAS DE  
MANUFATURA JOB-SHOP REALISTA COM DEMANDA POR  
ENCOMENDA: SOLUÇÃO POR ALGORITMO GENÉTICO**

**FLORIANÓPOLIS**

**2010**



ROGÉRIO MALTA BRANCO

**AGENDAMENTO DE TAREFAS EM SISTEMAS DE  
MANUFATURA JOB-SHOP REALISTA COM DEMANDA POR  
ENCOMENDA: SOLUÇÃO POR ALGORITMO GENÉTICO**

Tese de doutorado apresentada ao  
Programa de Pós-Graduação em  
Engenharia de Produção da  
Universidade Federal de Santa  
Catarina como requisito parcial para  
obtenção do grau de Doutor em  
Engenharia de Produção  
Orientador: Prof. Dr. Antônio S.  
Coelho.

FLORIANÓPOLIS

2010

Catálogo na fonte pela Biblioteca Universitária  
da  
Universidade Federal de Santa Catarina

B816a Branco, Rogério Malta

Agendamento de tarefas em sistemas de manufatura job-Shop realista com demanda por encomenda [tese] : solução por algoritmo genético / Rogério Malta Branco ; orientador, Antônio Sérgio Coelho. - Florianópolis, SC : 2010.  
1 v. : il., grafs., tabs.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Produção.

Inclui referências

1. Engenharia de produção. 2. Algoritmos genéticos. 3. Agendamento de tarefas. 4. Job-Shop. I. Coelho, Antônio Sérgio. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Produção. III. Título.

CDU 658.5

## **TERMO DE APROVAÇÃO**

### **AGENDAMENTO DE TAREFAS EM SISTEMAS DE MANUFATURA JOB-SHOP REALISTA COM DEMANDA POR ENCOMENDA: SOLUÇÃO POR ALGORITMO GENÉTICO**

por

**ROGÉRIO MALTA BRANCO**

Esta tese foi apresentada no dia 29 de março de 2010 como requisito parcial para a obtenção do título de DOUTOR EM ENGENHARIA DE PRODUÇÃO, Linha de Pesquisa – Logística e Transporte, Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Prof.<sup>a</sup>. Cátia Maria dos Santos Machado,  
Dr.<sup>a</sup>.  
(FURG)

---

Prof.<sup>a</sup>. Elaine Corrêa Pereira, Dr.<sup>a</sup>.  
(FURG)

---

Prof. José Leomar Todesco, Dr.

---

Prof. Sérgio Fernando Mayerle, Dr.

---

Prof.<sup>a</sup>. Mirian Buss Gonçalves, Dr.<sup>a</sup>.  
Presidente

---

Prof. Antônio Sérgio Coelho, Dr.  
Orientador

Visto da coordenação:

---

Prof. Antônio Cezar Bornia, Dr. Eng.  
Coordenador do Curso



Dedico cada pensamento, cada letra, cada idéia, ao meu filho Guilherme.  
Nunca deixa de sonhar e desejar teus sonhos...  
Eles se tornam realidade.





## AGRADECIMENTOS

A Deus, por tudo, sem exceções.

À minha família, razão de minha vida. A meu filho Guilherme, onde nos momentos de ausência sempre demonstrou seu amor incondicional com abraços gostosos, brincadeiras criativas e nos revigorantes “intervalinhos” nos dias de muito estudo. Aos meus pais, Dílson e Zair, que como sempre não mediram esforços para ajudar na realização do trabalho, sempre com carinho e palavras de fé. Aos meus amados irmãos Flávia e Dílson, e minha tia Zadir, por estarem sempre apoiando e perdendo a ausência. A Daniele, que companheira, sabe o quanto estes “vôos” me são necessários, cedeu do nosso tempo precioso pra que tudo desse certo.

Ao amigo e professor Antônio Sérgio Coelho, sempre mostrando que constância, parceria, paciência e amizade são fundamentais até para uma boa orientação de um trabalho.

Ao prof. Sérgio Mayerle, amigo, otimista, criativo e incansável pra ajudar sempre.

A todos os professores que se mostraram presentes, transmitindo seus conhecimentos durante o curso e desenvolvimento deste trabalho.

Aos colegas, pela amizade formada durante o curso, em especial aos amigos: Rosimeri, Roseli, Vanina, Camila e Hobbed, que contribuíram para a realização deste trabalho.

Aos amigos do IF-RS e UTFPR, que apoiaram a idéia, torceram e ajudaram de todas as formas possíveis para que este trabalho pudesse ser concluído.

Temos o “gene egoísta”, mas evoluímos também pela diversidade e coletividade. Ressalto minha emoção ao ver o quanto as pessoas se envolvem, apóiam, participam e vibram com o empenho de outra, muitas vezes mera desconhecida. Acredito que também por conta desta sinergia, o mundo evolui, e agradeço por poder ter vivido cada instante desta troca impar de experiências e aprendizado. Obrigado assim, a todos que, mesmo achando que pouco fizeram, ajudaram muito nesta jornada, que agora começa.



*“A alegria está na luta, na tentativa,  
no sofrimento envolvido e não na vitória  
propriamente dita.”*

*(Mahatma Gandhi, 1869-1948)*



## RESUMO

Diante de um mercado globalizado e de estruturas organizacionais cada vez mais complexas, o processo de tomada de decisão de uma empresa necessita, cada vez mais, ser rápido e eficiente a fim de promover uma boa posição estratégica. As incertezas de mercado e de produção levam as empresas a investir em flexibilidade na busca por maior robustez. Os Sistemas de Manufatura Flexíveis (SMFs) têm ganhado bastante espaço nas empresas, principalmente naquelas com produção por encomenda e em pequenos lotes, onde os clientes têm grande interação com a produção. Nesta ótica, o presente trabalho trata da programação da produção de sistemas de manufatura *Job-shop* envolvendo situações mais realistas. Aplicando-se um algoritmo genético (AG) que envolve regras de despacho combinadas, são selecionadas as rotas que apresentem os menores tempos de produção e de *setup*, sem descuidar do prazo de entrega. O AG proposto atua em duas frentes de forma conjunta: seleção dos recursos disponíveis na célula de manufatura para compor a rota e a ordem de execução de cada processo, dentro da célula. Para tanto é utilizado um cromossomo composto de duas partes: cabeça e corpo. As rotas advêm da primeira estrutura que combinadas com os índices de escolha dos recursos da segunda, refletirão nos agendamentos. Testes envolvendo casos clássicos e realistas são promovidos com base na literatura. Nos casos clássicos é observado o desempenho do sistema frente ao agendamento, dado não haver multiplicidade de rotas. Nos demais testes é observado o comportamento da proposta frente a situações com múltiplas rotas e quebras de recursos, bem como atrasos no início dos processos, alterações nos tempos de processamento e tempos dependentes do sequenciamento que está sendo adotado. Os resultados foram satisfatórios, observando-se que a proposta apresentada aproxima-se dos resultados ótimos para os problemas clássicos e, para os realistas, equiparou-se ou superou, segundo critérios de erro mínimo e tempo de processamento) os resultados da literatura, sob mesmas condições.

Palavras-chave: Algoritmos genéticos. Agendamento de tarefas. *Job-shop*.



## ABSTRACT

Due to a global market and more and more complex organizational structures, the decision making process in an enterprise needs, increasingly, to be quick and efficient in order to promote good strategic position. Production and market uncertainties cause companies to invest in flexibility in search for greater robustness. The Flexible Manufacturing Systems (FMSs) have grown considerably in companies, especially in those with production per order and in small lots, where customers have a great interaction with the production. In this perspective, this work deals with the scheduling production of Job-shop manufacturing systems involving more realistic situations. Applying a genetic algorithm (GA) involving combined dispatching rules, routes are selected with the lowest production times and setup, with special attention for delivery time. The proposed GA acts on two fronts jointly: selection of available resources in the manufacturing cell to compose the route and the order of execution of each process within the cell. For this purpose, a chromosome composed of two parts: head and body, is used. The routes derive from the first structure that, combined with the indices of choice of the second structure resources, will be reflected in the schedules. Tests involving realistic and classical cases are promoted based on the literature. In the classic cases, the system performance is observed against the scheduling, since there is no multiplicity of routes. In other tests, the proposal behavior is observed in face of multiple route situations and resource breaking, as well as delays in the beginning of the processes, changes in processing times and dependent times of the sequence that is being adopted. The results were satisfactory, noticing that the proposal gets close to the optimum results for classical problems and, for the realists, it was similar or surpassed, according to the criteria of minimum error and processing time, the results of the literature, under same conditions..

**Keywords:** Genetic algorithms. Scheduling tasks. Job-shop.





## LISTA DE FIGURAS

Figura 2.1 – Representação ampliada da função da produção .....	49
Figura 2.2 – Representação da função da produção sob a ótica da estratégia .....	50
Figura 2.3 – Visão geral das atividades do PCP .....	51
Figura 3.1 – Plano de processo com rotas alternativas nos subprocessos .....	67
Figura 3.2 –JSSP: (a) grafo disjuntivo; (b) <i>makespan</i> (C, D).....	70
Figura 3.3 – Solução de JSSP 3x3 – (a) gráfico de Gantt e (b) matriz sequência de tarefas .....	71
Figura 3.4 – Relação entre os tipos de schedules no JSSP .....	72
Figura 3.5 – Exemplo entre escalas semi-ativas e ativas.....	72
Figura 4.1 – Algoritmo de busca tabu, modificado para escalonamento de atividades .....	84
Figura 4.2 – Fluxograma simplificado de um algoritmo genético clássico .....	88
Figura 4.3 – Exemplo de codificação de soluções de JSSP em algoritmos genéticos.....	91
Figura 4.4 – Exemplo de roleta viciada para 20 indivíduos.....	94
Figura 4.5 – Exemplo de seleção por amostragem estocástica .....	96
Figura 4.6 – Exemplos: (a) ordenação de indivíduos; (b) proporções para roleta viciada .....	96
Figura 4.7 – Exemplos: (a) seleção de Mayerle (1994); (b) histograma dos selecionados para cruzamento .....	98
Figura 4.8 – Exemplo de operador de cruzamento de um ponto .....	99
Figura 4.8 – Exemplo de operador de cruzamento de dois pontos ....	100
Figura 4.8 – Exemplo de operador de cruzamento Uniforme.....	100

Figura 4.9 – Exemplo de geração de um dos dois filhos com operador de crossover OX.....	101
Figura 4.10 – Exemplo de geração de um dos dois filhos por crossover PMX.....	101
Figura 4.11 - Exemplo de geração de filho com operador de crossover PPX.....	102
Figura 4.12 – Exemplo de esquema alternativo de geração de populações.....	103
Figura 5.1 – Estrutura geral do cromossomo: cabeça e corpo .....	107
Figura 5.2 – Relação entre tabela original e cromossomo – fenotipo x genotipo.....	107
Figura 5.3 – Relação genótipo x fenótipo no seqüenciamento de operações .....	108
Figura 5.4 – Exemplo de Geração aleatória de “cabeça” de cromossomo .....	110
Figura 5.5 - estratégia para a formação das populações .....	118
Figura 6.1 – Tela do programa Genesis 2.32: configuração para cálculo .....	124
Figura 6.2 – Recortes de tela do programa Genesis 2.32: agendamento ativo para problema YN97 sem tempos de <i>setup</i> dependentes.....	144
Figura 6.3 – Recortes de tela do programa Genesis 2.32: Entrada de dados de tempos de <i>setup</i> dependentes .....	144
Figura 6.4 – Recortes de tela do programa Genesis 2.32: agendamento ativo para problema YN97 com efeito dos tempos de <i>setup</i> dependentes.....	146
Figura 6.5 –tela do programa Genesis: agendamento ativo ND para problema YN97 sem tempos de <i>setup</i> dependentes .....	147
Figura 6.6 –tela do programa Genesis: agendamento ativo ND para problema YN97 com tempos de <i>setup</i> dependentes .....	147

Figura 6.7 – Grafo representando agendamento ativo ND para problema YN97 (A) sem e (B) com tempos de <i>setup</i> dependentes .....	148
Figura 6.9 – Tela do programa Genesis: agendamento ativo ND do problema KUMAR (Kumar et al, 2003).....	151
Figura 6.10 – Tela do programa Genesis: agendamento ativo ND do problema KUMAR (Kumar et al, 2003) com quebra de máquina gargalo. ....	152
Figura 6.11 – Telas do programa Genesis com agendamentos ativo ND do problema Chan ( 2003): alocação de tarefas sem (A) e com (B) quebra na máquina gargalo e respectivos graficos de Gantt (C e D).....	154
Figura B.1 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo. ....	170
Figura B.2 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo. ....	171
Figura B.3 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo. ....	171
Figura B.4 – Gráfico relacionando erro % médio em relação ao ótimo. ....	172
Figura B.5 – Gráfico relacionando erro % mínimo em relação ao ótimo. ....	173
C. Dados da solução do problema clássico LA02.....	174
Figura C.1 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo. ....	175
Figura C.2 – Gráfico relacionando erro % médio em relação ao ótimo. ....	176
Figura C.3 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo. ....	176
Figura C.4 – Gráfico relacionando erro % médio em relação ao ótimo. ....	177

Figura C.5 – Gráfico relacionando erro % mínimo em relação ao ótimo.	178
Figura D.2 – Gráfico relacionando erro % médio em relação ao ótimo.	181
Figura D.3 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo.	181
Figura D.4 – Gráfico relacionando erro % médio em relação ao ótimo.	182
Figura D.5 – Gráfico relacionando erro % mínimo em relação ao ótimo.	183
Figura E.1 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo.	185
Figura E.2 – Gráfico relacionando erro % médio em relação ao ótimo.	186
Figura E.4 – Gráfico relacionando erro % médio em relação ao ótimo.	187
Figura E.5 – Gráfico relacionando erro % mínimo em relação ao ótimo.	188
Figura F.1 – Gráfico relacionando <i>makespan</i> médio e oscilações: heurísticas diversas; taxas PM de 1% e 5%; agendamentos Ativo e ativo ND.	190
Figura F.2 – Gráfico relacionando <i>makespan</i> médio e oscilações: heurísticas diversas; taxas PM de 1% e 5%; Agendamentos Ativo e ativo ND.	192
Figura G.1 – Gráfico relacionando <i>makespan</i> médio e oscilações: heurísticas diversas; taxas PM de 1% e 5%; agendamentos Ativo e ativo ND.	194
Figura G.2 – Gráfico relacionando tempos médios e oscilações: heurísticas diversas; taxas PM de 1% e 5%; agendamentos Ativo e ativo ND.	196

Figura H.1 – Gráfico relacionando <i>makespan</i> médio e oscilações: heurísticas diversas; taxas PM de 1% e 5%; agendamentos Ativo e ativo ND. ....	198
Figura H.2 – Gráfico relacionando tempo médio e oscilações, para diversas heurísticas, taxas PM de 1% e 5%, além dos agendamentos Ativo e ativo ND. ....	200



## LISTA DE QUADROS

Quadro 6.1 – Classificação dos 10 melhores (a) tempos médios E (b) erros médios do problema LA02: ordenação baseada no balanço tempo x erro.	125
Quadro 6.2 – Classificação dos 10 melhores (a) tempos médios E (b) erros médios do problema abz6: ordenação BASEADA no balanço tempo x erro	126
Quadro 6.3 – Classificação dos 10 melhores (a) tempos médios E (b) erros médios do problema FT10: ordenação BASEADA no balanço tempo x erro	127
Quadro 6.4 – Classificação dos 10 melhores (a) tempos médios E (b) erros médios do problema LA21: ordenação BASEADA no balanço tempo x erro	129
Quadro 6.5 – designação dos pontos para ordenação em função da ordem	131
Quadro 6.6 – Tempos de <i>setup</i> dependentes propostos para as três máquinas do YN97	146
Quadro B.1 – Resumo geral da solução do problema ABZ6: dados médios e mínimos	169
Quadro C.1 – Resumo geral da solução do problema LA02: dados médios e mínimos	174
Quadro D.1 – Resumo geral da solução do problema LA21: dados médios e mínimos	179
Quadro E.1 – Resumo geral da solução do problema FT10: dados médios e mínimos	184
Quadro F.1 – Resumo da solução do problema de CHAN original (CHAN,2003): <i>Makespans</i> médios e agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	189
Quadro F.2 – Resumo da solução do problema de CHAN original (CHAN,2003): tempos médios e agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	191

Quadro G.1 – Resumo da solução do problema CHAN modificado (ARAUJO,2006): <i>makespans</i> médios para agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	193
Quadro G.2 – Resumo da solução do problema CHAN modificado (ARAUJO,2006): tempos médios para agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	195
Quadro H.1 – Resumo da solução do problema Kumar et al (2003): <i>makespans</i> médios para agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	197
Quadro H.2 – Resumo da solução do problema Kumar et al (2003): tempos médios para agendamentos A e ND com oscilações de 0, 5, 10 e 20%.	199



## LISTA DE TABELAS

Tabela 3.1 - Variações de Máquinas do Tipo <i>Shop</i> (especializadas). ....	62
Tabela 3.2 - ALGUMAS características comumente associadas ao campo de tarefas. ....	63
Tabela 3.3 – Funções de custo mais comuns. ....	64
Tabela 3.4 – Problema JSSP 3x3 .....	71
Tabela 4.1 – Relação de algumas PDR's considerando datas devidas e tempo de fluxo.....	81
Tabela 4.2 – Relação de terminologias associadas aos AG's.....	89
Tabela 6.1 –JSSP clássicos: ordenação decrescente em função do tempo médio e média de pontos .....	132
Tabela 6.2 – JSSP clássicos: esquemas em ordem decrescente de erro médio e média de pontos .....	132
Tabela 6.2 – Tempos de processamento e rotas alternativas propostas por Chan (2003) .....	134
Tabela 6.3 – Tempos de processamento e rotas alternativas propostas por Chan (2003) modificado por Araujo (2006).....	135
Tabela 6.3 (cont.) – Tempos de processamento e rotas alternativas propostas por Chan (2003) modificado por Araujo (2006) .....	136
Tabela 6.4 – Tempos de processamento e rotas alternativas propostas por Kumar et al (2003) .....	138
Tabela 6.4 – Dados de um indivíduo formado aleatoriamente segundo informações do problema de agendamento de tarefas em job-shop 3x3 proposto por Yamada e Nakano (1997) .....	142
Tabela A.1 – Fontes originais dos trabalhos aplicados aos testes.....	166
Tabela A.2 – Valores ótimos dos problemas utilizados nos testes .....	166
Figura E.3 – Gráfico relacionando erro % médio e mínimo em relação ao ótimo.....	186



## LISTA DE SIMBOLOS

- A: conjunto das operações agendadas;
- C: conjunto das operações candidatas;
- $d_i$ : data devida do job  $i$ .
- $D_j$ : instante ou data de entrega da tarefa  $j$ ;
- $d_{kj}$ : data de entrega da operação  $k$  na tarefa  $j$ ;
- $E_j$ : instante de chegada da tarefa  $j$  ao sistema;
- G: conjunto das operações conflito;
- $i$ : operação;
- $j$ : tarefa (job);
- $k$ :  $k$ -ésima operação da tarefa  $j$ ;
- K: tamanho da população;
- M: número total de máquinas para realizar as tarefas, em um cromossomo;
- $M'$ : número total de operações de cada tarefa;
- $M_i$ : número de operações que faltam para o job $_i$  finalizar-se;
- $M_q$ : número da máquina que sofreu quebra;
- N: número de tarefas;
- $o_{ik}$ : operação  $k$  do job  $i$ ;
- $o_{ike}$ : operação extra ou fragmento de  $o_{ik}$  até instante da quebra
- $o_{ikm}$ : operação manutenção;
- $o_{ikq}$ : operação alocada na máquina no instante da quebra;
- $p_{ij}$ : tempo de processamento da operação  $i$  da tarefa  $j$ ;
- $P_{ijq}$ : tempo de duração da quebra/manutenção na máquina;
- $p_{ik}$ : tempo de processamento da operação  $k$  da tarefa  $i$ ;
- $p_m$ : probabilidade de ocorrência de mutação;
- $r$ : tempo requerido pelas oper. restantes de  $j$ , a partir de  $t$ , considerando  $p$ ;
- $s'$ : folga da operação, sendo  $s' = D - c.(r - p)$ , sendo  $c$  = parâmetro;
- $s_j$ : folga da tarefa  $j$  ( $s = D - t - r$ );
- $t$ : instante do tempo em que uma operação é alocada a um recurso produtivo;
- $t_{ik}$ : tempo apto a iniciar a operação  $k$  do job  $i$ ;
- $T_{iq}$ : tempo de início da quebra/manutenção;
- $t_p$ : média do tempo de processamento da operação, na máquina;
- $t_p'$ : tempo total de processamento da operação, na próxima máquina;
- $\alpha|\beta|\gamma$ : notação de Bruker para a definição de um problema de escalonamento;



## **LISTA DE ABREVIATURAS EM PORTUGUES**

AG: Algoritmo genético  
MC: Manufatura celular;  
NP: Não polinomial;  
RN: Regra de despacho randômica (aleatória);  
RHC: Regra heurística combinada;  
SM: Sistema de manufatura;  
SMF: Sistema de manufatura flexível;  
SP: Sistema produtivo;  
CNC: Comando numérico computadorizado;  
PCP: Planejamento e controle da produção;  
CM: Célula de manufatura;  
PMP: Plano-mestre de produção;  
CQ: Controle de qualidade;



## LISTA DE TERMOS ESTRANGEIROS

ASSEMBLE TO ORDER:	montagem por encomenda;
BATCH:	lote, porção;
CHAN:	problema proposto por Chan (2003);
GT:	algoritmo de agendamento proposto por Giffler e Thompson (1960);
KUMAR:	problema proposto por Kumar (2003);
LEAD TIMES	é o tempo entre o momento da entrada material e a saída do inventário. Relaciona-se a produto ou atividade;
LEAN PRODUCTION	produção enxuta;
LPT:	<i>longest processing time</i> – regra heurística envolvendo maior tempo de processamento;
MAKE TO ORDER:	produção sob encomenda;
S/RPT:	<i>slack per remaining time</i> - regra heurística envolvendo folga em função do prazo de entrega, andamento do processo e tempo necessário para finalizar;
SCHEDULING:	agendamento, escalonamento;
SPT:	<i>shortest processing time</i> – regra heurística envolvendo menor tempo de processamento;
YN97:	problema proposto por Yamada e Nakano (1997);





# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>37</b>
1.1 OBJETIVOS .....	41
1.1.1 <i>Objetivo geral</i> .....	41
1.1.2 <i>Objetivos específicos</i> .....	41
1.2 JUSTIFICATIVA E RELEVÂNCIA DO TEMA .....	42
1.3 LIMITAÇÕES DO TRABALHO .....	44
1.4 CONTRIBUIÇÕES DO TRABALHO .....	45
1.5 ESTRUTURA DO TRABALHO .....	46
<b>2 SISTEMAS DE PRODUÇÃO.....</b>	<b>48</b>
2.1 ASPECTOS GERAIS .....	48
2.2 PLANEJAMENTO E CONTROLE DA PRODUÇÃO (PCP) .....	51
2.3 PROGRAMAÇÃO DA PRODUÇÃO E SEQUENCIAMENTO DE TAREFAS .....	53
2.4 CLASSIFICAÇÕES DOS SISTEMAS PRODUTIVOS .....	54
2.5 JOB-SHOP COM PEQUENOS LOTES E PRODUÇÃO POR DEMANDA .....	57
<b>3. ESCALONAMENTO EM SISTEMAS DE PRODUÇÃO .....</b>	<b>59</b>
3.1 AGENDAMENTO DE TAREFAS .....	59
3.2 CLASSES DO PROBLEMA DE AGENDAMENTO DE TAREFAS .....	61
3.2.1 <i>Notação de Bruker</i> .....	61
3.2.2 <i>Casos de JSSP clássicos e realistas</i> .....	65
3.2.3 <i>Casos reais: características e estratégias envolvidas</i> .....	66
3.3 AS FORMAS DE REPRESENTAÇÃO DOS JSSP .....	69
3.4. AS CLASSES DE AGENDAMENTO PARA O JSSP .....	71
<b>4. TÉCNICAS APLICADAS NA SOLUÇÃO DOS JSSP .....</b>	<b>73</b>
4.1. TÉCNICAS EXATAS DE SOLUÇÃO .....	73
4.2. TÉCNICAS DE SOLUÇÃO APROXIMADAS .....	75
4.2.1. <i>Regras de despacho</i> .....	76
4.2.2. <i>Sistemas de base de conhecimento</i> .....	82
4.2.3. <i>Relaxação lagrangeana</i> .....	82
4.2.4. <i>Metaheurísticas</i> .....	83
4.3 ALGORITMOS GENÉTICOS APLICADOS AO JSSP .....	87
4.3.1. <i>Terminologia</i> .....	88

4.3.2. A codificação das soluções: os cromossomos .....	89
4.3.3. A população inicial.....	91
4.3.4. A função de avaliação.....	92
4.3.5. O processo de seleção.....	93
4.3.6. Os operadores genéticos.....	98
<b>5. MÉTODOS APLICADOS AO PROBLEMA DE AGENDAMENTO .....</b>	<b>105</b>
5.1 CLASSIFICAÇÃO METODOLÓGICA DA PESQUISA.....	105
5.2 SOLUÇÃO DO JSSP PROPOSTO NO TRABALHO.....	106
5.2.1 Estrutura do cromossomo proposto .....	106
5.2.2. Formação da população inicial.....	109
5.2.3. Função aptidão e avaliação dos indivíduos .....	110
5.2.4. A seleção de indivíduos.....	114
5.2.5. Os operadores genéticos.....	115
5.2.6. Os parâmetros do AG.....	117
5.3 REGRA HEURÍSTICA COMBINADA – RHC .....	119
<b>6. APLICAÇÃO DA PROPOSTA DE AGENDAMENTO E ANÁLISE DOS RESULTADOS: TESTES PARA CASOS CLÁSSICOS E REALISTAS.....</b>	<b>121</b>
6.1 TESTES DA ETAPA AGENDAMENTO – CASOS CLÁSSICOS DE JSSP .....	122
6.1.1 Métodos adotados para os testes.....	122
6.1.2 Resultados dos testes de agendamento clássico .....	124
6.2 TESTES ENVOLVENDO AUMENTO DE ROTAS ALTERNATIVAS.....	133
6.3 TESTES ENVOLVENDO JSSPS DINÂMICOS .....	137
6.3.1 Resultados e discussões .....	139
6.3.1.2 Problema CHAN modificado .....	140
6.3.1.3 Problema Kumar .....	140
6.4 TESTES ENVOLVENDO AGENDAMENTO COM TEMPOS DE <i>SETUP</i> DEPENDENTES	141
6.4.1. Agendamento com tempos de <i>setup</i> dependentes: demonstrando o funcionamento do algoritmo GT e genético.....	141
6.4.2. Agendamento em formação ND considerando diversos tempos de <i>setup</i> dependentes .....	146
6.5 TESTES ENVOLVENDO AGENDAMENTO EM PROBLEMAS COM QUEBRAS DE MÁQUINAS	148
6.5.1 – Teste de quebra de máquina: problema KUMAR.....	150
6.5.2. Teste de quebra de máquina: problema CHAN original .....	152
<b>7. CONCLUSÕES FINAIS .....</b>	<b>155</b>

7.1 CONTRIBUIÇÕES .....	155
7.2 TRABALHOS FUTUROS .....	156
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>159</b>
A. PROBLEMAS CLÁSSICOS DE AGENDAMENTO JOB-SHOP UTILIZADOS .....	166
A.1. <i>Problema ABZ6</i> <sub>[10 x 10]</sub> .....	167
A.2. <i>Problema LA02</i> <sub>[10 x 5]</sub> .....	167
A.3. <i>Problema LA21</i> <sub>[15 x 10]</sub> .....	167
A.4. <i>Problema MT10</i> <sub>[10 x 10]</sub> (ou FT100).....	168
B. DADOS DA SOLUÇÃO DO PROBLEMA CLÁSSICO ABZ6 .....	169
E. DADOS DA SOLUÇÃO DO PROBLEMA CLÁSSICO FT10.....	184
F. DADOS DA SOLUÇÃO DO PROBLEMA COM ROTAS ALTERNATIVAS – CHAN (CHAN,2003).....	188
G. DADOS DA SOLUÇÃO DO PROBLEMA COM ROTAS ALTERNATIVAS – CHAN EXPANDIDO (ARAUJO, 2006).....	193
H. DADOS DA SOLUÇÃO DO PROBLEMA COM ROTAS ALTERNATIVAS – KUMAR (KUMAR ET AL, 2003).....	197



## 1. INTRODUÇÃO

A crescente competição nas indústrias provocada pela globalização do mercado tem atraído a atenção das empresas para a qualidade e a produtividade. Não são esforços recentes, pois desde os primeiros sinais de ineficiência do modelo taylorista-fordista, ainda nos anos 60, que as linhas de produção especializadas e fabricando grandes lotes de poucas variedades de peças tem cedido lugar a sistemas mais flexíveis, produzindo lotes mínimos e com grande variedade. Esta inversão na ótica da produção e da organização da manufatura advém da busca por melhor posicionamento estratégico, onde a flexibilidade é a chave para garantir maior agilidade e eficiência na manufatura.

Neste contexto têm-se os sistemas de manufatura do tipo *Job-shop*, caracterizados por apresentarem baixos volumes de produção e elevada variedade de produtos. Neles, organizar o processamento de tarefas em recursos flexíveis é um problema bastante complexo e conhecido como problema de escalonamento de tarefas (ou JSSPs – *Job-shop scheduling problem*). Em geral, dado um grupo de tarefas e outro de recursos, consiste em alocar temporalmente as operações de cada tarefa aos recursos disponíveis. As tarefas são seqüências de operações com tempos ininterruptos de processamento para cada máquina.

Havendo maiores anseios por máquinas flexíveis, executando diferentes tarefas e rotas flexíveis (caminhos diferentes para produzir as mesmas tarefas), estes problemas de escalonamento ficam ainda mais complexos, contudo estas possibilidades abrem caminho a possíveis vantagens produtivas e competitivas.

Historicamente, a busca por flexibilidade de máquinas teve grande salto com o as máquinas-ferramentas com Comando Numérico Computadorizado (CNC), ao aliar flexibilidade com automação, antes presente apenas em sistemas rígidos de linhas de produção especializadas. Esta troca de automatização pela automação de maquinário trouxe consigo a possibilidade de fabricar lotes de produtos variados, dando ao sistema de manufatura maior capacidade de adaptação frente a uma demanda instável de quantidade e qualidade.

Oriundas das possibilidades que o CNC ampliou às máquinas-ferramentas, novos arranjos tecnológicos e organizacionais são possíveis. Dimopoulos e Mort (2001), consideram que o passo inicial para um sistema flexível de manufatura (SMF) é a Manufatura Celular (MC), alcançada através da organização da manufatura em um *layout* celular. Fruto de metodologias como a Tecnologia de Grupo (TG), que

conceituada por Mitrofanov no trabalho “*Scientific Principle of Group Technology*”, consiste inicialmente em agrupar peças com similaridade geométrica em famílias para fabricação em um tipo de máquina.

Como aprimoramento das células de manufatura (CM), os SMFs fabricam as famílias de peças, contudo visando menores tempos de atravessamento e grande flexibilidade na produção das diferentes peças que compõe a família. Em linhas gerais, são um conjunto de máquinas CNC, sistemas de transporte automático entre estações e postos de trabalho, além de um computador que controla todo o funcionamento.

De acordo com Goldhar (1984), os SMFs podem conduzir a grandes vantagens econômicas, além do fato de que, para empresas que irão disputar no mercado, seja doméstico ou externo, torna-se fundamental automatizar suas operações de manufatura visto que países como Japão e Alemanha o tem feito vigorosamente.

Com as empresas investindo cada vez mais em aprimoramentos organizacionais e tecnológicos, é fácil ver que um menor tempo de resposta para as tomadas de decisão gerenciais é crucial na definição de uma vantagem competitiva. Os sistemas baseados na Tecnologia da Informação (TI) estão sendo mais frequentemente empregados visto que auxiliam de forma impar a elevar a agilidade e confiabilidade destas decisões gerenciais. Neste âmbito, o PCP – Planejamento e Controle da Produção surge como sistema integrado de administração da produção. Fundamental no processo decisório da manufatura tem na programação das operações seu maior desafio: organizar os recursos produtivos de forma a definir quando e como produzir.

Este seqüenciamento de tarefas é uma etapa de curto prazo do PCP, e constitui problemas de agendamento de produção, que são extensamente estudados nas áreas de Pesquisa Operacional (PO), ciências da computação e inteligência artificial, objetivando encontrar a melhor ordem de processamento de uma sequencia de tarefas para que se tenha o processamento mais eficiente – menor tempo total de processamento (*makespan*), por exemplo.

Dadas às dimensões da estrutura disponível (recursos e peças), assume condições difíceis de solucionar em tempo polinomial. Isto é posto em virtude da natureza combinatória do problema de alocar máquinas à produção das peças. Somando-se o interesse em flexibilidade de rotas de fabricação, tem-se um aumento nos recursos disponíveis, o que aumenta ainda mais as dimensões do problema, tornando-o ainda mais complexo.

Entretanto, técnicas heurísticas são também muito pesquisadas e aplicadas em problemas deste tipo, onde apesar de não garantirem a

otimalidade, permitem obter bons resultados em tempos relativamente baixos, o que é bastante satisfatório. O maior desafio neste caso reside em escolher a melhor forma de solucionar o agendamento, ou seja, em escolher o foco para buscar a eficiência. Usualmente o *makespan* é uma função objetivo bastante aplicada para casos clássicos de agendamento, onde somente o tempo máximo de finalização dos processos é interessante. Quando se obtêm modelos menos restritos, como nos casos reais dos sistemas de produção, outros fatores tornam-se muito mais interessantes, como por exemplo, o tempo em atraso de um processo.

Preocupações com tempos em atraso são frequentes em sistemas produtivos que apresentam forte interação junto aos clientes/parceiros. Isto tem se tornado importante na definição das vantagens competitivas.

Associando os termos: flexibilidade, agilidade e qualidade, observa-se tendências recentes envolvendo as SMFs e a produção enxuta. O *Just-in-time* evidencia-se nestas discussões, como principal pilar da produção enxuta a ser integrado aos SMFs.

O *Just-in-time* é, segundo Shonberger (1984) um conceito que pode ser descrito como a produção de uma unidade de um produto para ser incorporada na hora certa em um processo subsequente. Em outras palavras, esta administração da produção determina que nada deve ser produzido, transportado ou comprado antes da hora exata. Isto implica na redução de estoques e dos custos decorrentes, com insumos/produtos chegando ao local de produção no exato instante de processamento.

Observa-se que a não disponibilidade de estoques leva o sistema a certas vulnerabilidades. Como os produtos são fabricados conforme produção por demanda - *make-to-order* (vende-se o produto para depois comprar a matéria prima e posteriormente fabricá-lo ou montá-lo), os lotes movimentados são sempre muito pequenos e o grau de credibilidade dos fornecedores deve ser elevado a fim de assegurar a qualidade e confiabilidade do fornecimento (Cheng et. al., 1996, p. 106).

Tocante ao tipo de manufatura, sendo o *layout* celular a base para um SFM, os baixos volumes de produção (tendendo a serem unitários) e ampla variedade de produtos caracterizam, segundo Porter et al (1999), as do tipo *Job-shop*, onde a alta qualificação é necessária para os profissionais que operam estes sistemas flexíveis de produção.

Os *job-shops* são sistemas de natureza combinatória, onde dada a flexibilidade dos recursos, as tarefas designadas a estes e a necessidade de seguir uma sequência tecnológica define um problema complexo, com difícil solução em tempo polinomial, consistindo na alocação temporal das operações de cada tarefa aos recursos disponíveis. As tarefas se dividem em sequências de operações, cada qual de tempo

ininterrupto de processamento, em uma dada máquina.

Sobre esta ótica o presente trabalho vem propor o agendamento de tarefas em sistemas de manufatura *Job-shop*. Empregando técnicas heurísticas combinadas e algoritmo genético, considera instabilidades que permeiam os sistemas reais de produção na formação dos seqüenciamentos (p.ex.: quebras de máquinas, atrasos, etc.).

Atuando sobre SMFs e células virtuais, os agendamentos promovem a escolha da rota através da função multiobjetivo que alia o menor tempo de processamento total, penaliza os atrasos no processamento e associa os tempos de *setup* dependentes.



## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Apresentar uma técnica de solução de problemas de agendamento dinâmico de tarefas em sistemas de manufatura da classe *job-shop*, considerando aspectos reais de produção e entrega das tarefas nos prazos devidos e com menor tempo de processamento, por meio de técnica evolutiva (algoritmos genéticos) e regra heurística combinada.

### 1.1.2 Objetivos específicos

Os objetivos específicos do trabalho são:

a) apresentar um algoritmo de escalonamento eficiente baseado em regra heurística combinada e contemplando os objetivos de menor tempo de processamento e entrega em datas devidas. Integrado a esta prática de escalonamento, é fim obter uma regra heurística combinada eficiente para alocar tarefas aos recursos ativos no horizonte de planejamento, considerando quebras de máquinas, atrasos nos tempos de início, variações nos tempos de processamento das tarefas e tempos de *setup* dependentes;

b) apresentar um algoritmo genético eficiente para formar, tanto rotas quanto sequenciamentos em cada rota, integrando o método de agendamento proposto e atuando com bons tempos computacionais de convergência e estratégias, tanto de formação de população inicial como de seleção e inserção de indivíduos em populações correntes;

c) desenvolver um *software* que permita integrar os algoritmos de agendamento e genético, promovendo o cadastro de: equipamentos, tempos de *setup* entre recursos, tempos de processamento médio em cada recurso, tempos máximos de entrega, além de definir parâmetros e estratégias referentes aos algoritmos evolutivos;

d) avaliar o comportamento dos sistemas de JSSP dinâmicos mediante submissão de exemplos colhidos da literatura, que agendados pelo sistema proposto permitirão tecer análises referentes à eficiência dos algoritmos propostos ante situações dinâmicas de escalonamento e uso de recursos;

## 1.2 JUSTIFICATIVA E RELEVÂNCIA DO TEMA

As exigências emergentes de um mercado globalizado influenciam diretamente nas buscas por melhores desempenhos, onde a flexibilidade tem sido a chave para a adaptação das organizações a este ambiente altamente competitivo. A necessidade de observar o consumidor, o fornecedor, o sistema financeiro, a tecnologia, as concorrências, mão-de-obra e tantos outros elementos que permeiam a empresa, vem por definir o caráter sistêmico do problema, sendo bastante complexo lidar com inúmeros fatores que agem simultaneamente sobre as empresas.

A necessidade de flexibilidade reflete diretamente na manufatura, levada a primar pela redução de estoques, tempos de atravessamento (*lead-times*), dentre outros. A necessidade do planejamento e controle de produção é fundamental no apoio à manufatura e, por conseguinte, de vantagens competitivas necessárias para o sucesso da empresa.

Também a importância que o cliente tem na definição da produção, hoje cada vez mais específica e com lotes quase unitários, o sequenciamento de tarefas, no curto-prazo do planejamento e controle da produção, necessita ser estável e ágil, dando suporte efetivo para o processamento das tarefas e considerando as exigências definidas pelo cliente.

A programação da produção nos ambientes de manufatura realistas da classe *job-shop*, tema desta tese, é bastante complexa para ser solucionada em tempo polinomial razoável, logo, soluções ótimas são consideravelmente difíceis de serem alcançadas. Observa-se que ambientes reais também não almejam tal preciosismo, até porque muitos dos dados determinísticos ligados às atividades realizadas nos ambientes fabris apresentam deteriorações ao longo do tempo causadas pelas incertezas intrínsecas dos processos (ZHOU, FENG e HAN, 2001). Assim, o tratamento destes problemas deve considerar modelos dinâmicos, e se possível, terem observância quanto aos não-determinismos inerentes de cada situação.

Apesar de existirem vários trabalhos abordando o tema de agendamentos em *job-shops*, Fernandes (1999) e Moccellini e Berllot (1998) apud Landmann (2005) salientam que a área de sequenciamento de tarefas, dentro da programação da produção, continua sendo carente de desenvolvimento e contribuições.

Assim, observa-se que a justificativa deste trabalho está alicerçada sobre a tendência mundial de melhoras no campo tático e operacional buscando melhor posicionamento estratégico. Com o PCP

na administração e controle da produção, ainda muitos esforços são aplicados no âmbito operacional, e o tema desta tese foca essencialmente a programação da produção, etapa de curto prazo do PCP, para propor incrementos nos índices de eficiência quanto aos escalonamentos e despachos de operações.

Quanto a operar com lotes pequeno-unitários, Araujo (2006) salienta que 75% das peças produzidas nas empresas do setor metalúrgico são em lotes inferiores a 50 peças, havendo tendência de elevação da diversidade destas peças e de produtos, aumentando conseqüentemente a necessidade de melhores resultados nas técnicas envolvidas no planejamento e controle da produção. Também Lucero (2001) afirma que a manufatura voltada para estes lotes é, dentre os processos produtivos existentes, a menos produtivos. Assim, estas condições justificam as ações de pesquisa sobre o tema, motivando na contribuição com métodos e técnicas eficientes no escalonamento e despacho de tarefas.

Quanto à complexidade expressa pelo problema de agendamentos em *job-shops*, o uso de técnicas de otimização como as metas-heurísticas é justificado, principalmente por serem excelentes ferramentas de busca por soluções. Definidos os algoritmos genéticos como sendo a ferramenta a ser aplicada, esta meta-heurística apresenta a robustez necessária para atuar em diferentes problemas realistas sem necessitar de extensivos ajustes de parâmetros, muitas vezes relacionados a cada caso específico de aplicação. Com o intuito de dispor de métodos que se apliquem a vários casos reais, também é justificada a escolha dos AGs como ferramenta de busca.

Como as meta-heurísticas permitem a inserção de heurísticas diversas em seu núcleo, aplicáveis a cada passo, escolher apropriadamente a forma como dispor destas heurísticas é fator motivador para a tese, principalmente por operar conjuntamente com um agendador que também permite variações heurísticas. Estas variações podem permitir obter aprimoramentos nos resultados focando cada objetivo definido para *schedule*.

Com a proposta de dispor de métodos aplicados ao problema de *job-shops* realistas fica evidenciado o interesse por este estudo, dadas às carências aqui evidenciadas. Soma-se também a motivação de apresentar ferramentas para sistemas focados nos clientes, considerando resposta de demanda *make-to-order*, preocupação de muitas empresas atuantes no cenário mundial, movidas pela necessidade crescente de agilidade. Não obstante, propor métodos que considerem os casos reais é outra grande motivação, haja visto a severa dificuldade em definir

características comuns a diversos casos práticos e também de modelá-las de forma a integrar técnicas de solução baseadas em métodos aproximativos que venham a culminar em resultados gerais com bons índices de eficiência – qualidade das soluções e tempo computacional investido para tanto. De maneira geral, considera-se também a motivação de poder aplicar o potencial sempre emergente da informática, não apenas como meio de registro, mas como ferramenta para auxiliar na aplicação das técnicas propostas pelo trabalho, tanto na construção, como análise e aprimoramento dos resultados.

### 1.3 LIMITAÇÕES DO TRABALHO

O agendamento de tarefas em ambientes *job-shop* é um tema amplamente discutido na literatura por mais de 50 anos, onde a comunidade de pesquisa operacional tem devotado grandes esforços na busca de métodos eficientes para solucionar este tipo de problema de natureza combinatória. Técnicas meta-heurísticas como *Simulated annealing*, *Busca tabu*, *Algoritmos genéticos*, dentre outras, têm sido aplicadas nos problemas complexos de otimização combinatória. Entretanto, tais aplicações estão envoltas, em geral, a casos clássicos do problema, ou seja, aproximações bastante reduzidas do problema encontrado em ambientes reais, onde restrições mais complexas devem ser consideradas.

Tocante ao foco do trabalho, os JSSP's dinâmicos reais, mesmo definidas as características específicas de aplicação para pequenos lotes/lotes individuais e atendendo a demandas *make-to-order*, ainda assim é um assunto amplo. São necessárias algumas limitações a fim de favorecer as discussões que envolvem a pesquisa. Até o momento são observadas as seguintes:

a) quanto ao escopo – o trabalho se concentra na construção de um modelo para solucionar problemas de escalonamento de operações em sistemas de manufatura da classe *job-shop*, com resposta a demanda do tipo *make-to-order*, dinâmico e voltado à produção de lotes pequenos e mesmo unitários (quebras de lotes e overlaps são desconsiderados). Basicamente, este modelo atua como parte integrante do PCP, a nível de curto-prazo, compartilhando dados do sistema para seu funcionamento, como por exemplo, horários de disponibilidade, capacidade de recursos, agenda de manutenção de máquinas, disponibilidade de material, etc.

b) quanto ao problema e generalização de resultados – o modelo

proposto se limita ao desenvolvimento de algoritmos genéticos e de agendamento. Os primeiros atuam sobre a formação de seqüências de operações e recursos disponíveis, dentre as rotas alternativas disponíveis, mediante atuação dos algoritmos genéticos. Os de agendamento, mediante aplicação de regras heurísticas combinadas atuam na formação e decodificação de soluções, voltando-se segundo estratégia selecionada, a agendamentos ativos ou ativos sem atraso. Numa fase posterior, de aplicação dos algoritmos em problemas com características reais, tarefas poderão ser simuladas e agendadas a recursos previamente definidos. Tempos de *setup* e de processamento podem sofrer oscilações por meio de curvas de distribuição de probabilidade. Considerando que os estudos compõem tanto avaliações de estratégia construtiva de populações iniciais, de seleção e cruzamento, de agendamento ativos e/ou ativos sem atraso além da definição, projeto e aplicação das RHC's, os resultados obtidos destes estudos poderão compor bases para agendamentos em categorias específicas dos JSSP's reais dinâmicos.

c) quanto à coleta de informações – como não é explícita a aplicação específica do ambiente real de manufatura JSSP dinâmico estudado (estudo de caso, por ex.), parte-se por limitar o problema quanto às características mais comumente encontradas em ambientes *job-shop* reais, conforme consultas literárias. Isto não impede a aplicação deste trabalho em um caso específico, bastando contudo, possíveis alterações necessárias na adaptação das restrições de aplicação.

## 1.4 CONTRIBUIÇÕES DO TRABALHO

Esta tese visa contribuir nos aspectos tocantes a manufatura *job-shop*, especificamente ao considerar aspectos realistas no agendamento de tarefas aliado a produção em lotes pequenos e unitários e resposta à demanda voltada ao cliente como objeto. Considerando tratar-se de hipóteses de trabalho, tem-se, contudo, que as principais contribuições da tese residem nos seguintes pontos:

- definição de um algoritmo construtivo híbrido para a geração de soluções de escalonamento, que considerando aleatoriedade e gulosidade, venha a permitir uma convergência mais eficiente ao sistema como um todo;
- definição de um algoritmo eficiente de geração de

agendamentos ativos e ativos sem atraso sobre rotas alternativas de processo, mediante aplicação de regras heurísticas combinadas;

- definição de uma estratégia de seleção de indivíduos do algoritmo genético que combine técnicas elitistas e probabilísticas visando maior robustez do algoritmo genético e conseqüentemente do sistema;

- definição de um algoritmo genético que integre a etapa de geração de escalas e as funções de avaliação, promovendo maior robustez ao sistema;

- considerar a possibilidade de escassez de rotas em dado instante e com isso reduzir-se o problema a forma clássica.

## 1.5 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em outros 6 capítulos, além desta introdução.

No capítulo introdutório é contextualizada a importância da problemática discutida nesta tese tocante a busca por vantagens competitivas e a sua relação com o escalonamento de tarefas em ambientes de manufatura. Conceitos de manufatura enxuta e sua integração com os sistemas flexíveis no aprimoramento da eficiência e agilidade dos sistemas produtivos são os motivadores deste trabalho e assim, também postos em pauta. Na sequência são expostos os objetivos, justificativas, limitações e contribuições do trabalho.

O segundo capítulo contextualiza os sistemas produtivos: definição, classificação frente algumas características de produção e demanda, bem como se discute o Planejamento e Controle da Produção (PCP) como ferramenta eficiente de gerenciamento de suas variáveis. Na classificação dos sistemas produtivos, caracterizam-se os *job-shops* e a produção por encomenda. No tocante ao PCP, discute-se sua atuação nos campos estratégico, tático e principalmente no operacional, onde se localiza a programação da produção e o seqüenciamento de tarefas.

O terceiro capítulo promove uma revisão da literatura envolvendo o agendamento de tarefas, definições, tipos e classes de agendamento. Na sequência, tocante as formas de solução, o capítulo 4 apresenta as principais técnicas envolvidas na busca por soluções eficientes, traçando as principais diferenças entre o problema clássico e realista, tanto relativo à formulação quanto às formas de solução mais evidentes.

Também no capítulo 4 são abordados com mais afinco o uso dos algoritmos genéticos na solução de problemas de *job-shop*. Partindo da sua terminologia, discute-se as formas de codificação adequadas para

problemas combinatórios e, para cada etapa deste algoritmo evolutivo são tratadas as principais estruturas normalmente empregadas, elucidando aquelas com maior eficiência na aplicabilidade tendo em vista o problema em estudo.

Da definição do problema, do conhecimento de formas de solução em uso e da motivação em propor um método eficiente para auxiliar na programação da produção, o quinto capítulo trata dos métodos a serem aplicados para testar a eficiência dos algoritmos propostos nesta tese, integrando os pontos evidenciados nos capítulos a montante.

No sexto capítulo são apresentados os testes realizados para validar a proposta desta tese, sendo definida uma metodologia de aplicação dos testes e, posteriormente, descritas as aplicações e resultados. Também são tecidas as conclusões sobre os resultados obtidos para cada teste.

No sétimo capítulo trata-se das limitações do trabalho e das contribuições futuras.

Ao final são apresentadas as referências utilizadas neste trabalho, seguidas dos Apêndices.

## 2 SISTEMAS DE PRODUÇÃO

Este capítulo faz referência aos sistemas de produção e sua importância na organização da produção. Discute-se ainda a importância da administração da produção frente à concorrência e a gerência de variáveis oriundas dos ambientes interno e externo da empresa na busca por vantagens competitivas.

### 2.1 ASPECTOS GERAIS

Até meados do século XVIII, a produção era basicamente familiar e o artesão realizava todas as atividades e detinha os meios de produção: instalações, ferramentas e matéria-prima. Com a Revolução Industrial desencadeando um aumento no consumo, houve a substituição do modo de produção doméstico pelo sistema fabril e assim, a necessidade da aplicação dos sistemas de produção (SP) na organização do trabalho.

Definido como um conjunto de recursos humanos, físicos, tecnológicos e informacionais, um sistema de produção pode ser capaz de transformar entradas em saídas, tangíveis ou não, possibilitando obter tanto bens como serviços. (SPRAKEL e SEVERIANO FILHO, 1999).

Dada a amplitude de suas funções operacionais, Tubino (2000) as organiza em três grupos funcionais básicos: finanças, com a administração dos recursos financeiros da empresa e sua alocação onde forem necessários; produção, que é qualquer atividade relacionada diretamente à transformação, mediante um ou mais processos de conversão (de insumos a bens ou serviços) e agregam valor aos insumos; marketing, promove e vende os bens e serviços produzidos pela empresa, tomando decisões sobre estratégias de publicidade e estimativas de preços.

Endossando a necessidade da forte inter-relação entre eles no compartilhamento de informações, fundamental para um desempenho eficiente do sistema, contudo, o foco principal do trabalho está sobre as questões relacionadas à função produção. Na Figura 1.1, tem-se sua representação de forma mais ampliada, onde não sendo tão restrita ao processo de transformação, relaciona também operações de apoio à produção, como o controle de qualidade (CQ), a manutenção, a logística e o Planejamento e Controle de Produção (PCP). Também ilustra outras saídas do processo de transformação, como sub-produtos, refugos, etc.





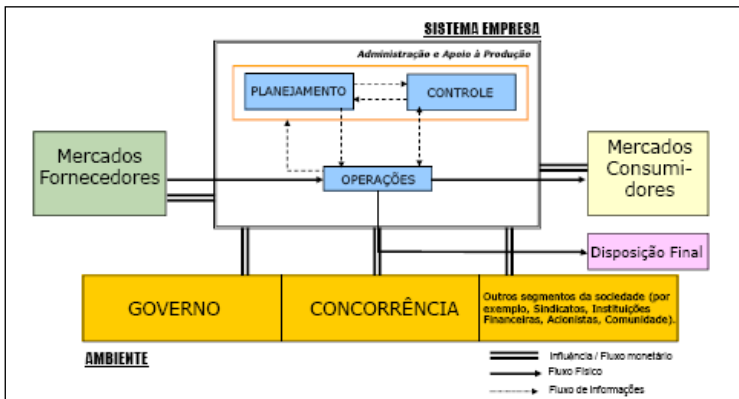
FONTE: FRANCISCHINI, 2007

FIGURA 2.1 – REPRESENTAÇÃO AMPLIADA DA FUNÇÃO DA PRODUÇÃO

Discutidos alguns dos principais fatores que levam a busca por melhor posicionamento estratégico, é trivial que se busque maior eficiência dos sistemas produtivos. Isto reflete, obviamente, em suas funções de produção, que associando insumos, recursos e tecnologia, necessitam ter seus resultados maximizados.

Sensível a influências diretas tanto dos ambientes interno quanto externo, o problema reside em como operar com estas variáveis para tirar máximo proveito do SP. Oishi (1995) apud Landman (2005) evidenciam esta questão não tão simples de resolver. Neste ponto é que surge a administração da produção como elemento a proporcionar esta busca por eficiência, ou seja, maior produtividade. Cabe salientar que esta administração está fortemente ligada aos interesses estratégicos da empresa que detém o sistema produtivo, considerando seus interesses e necessidades junto ao mercado.

Na Figura 2.2 Francischini (2007) apresenta este envolvimento, onde se observa a influência das variáveis ambientais no SP, especificamente na função produção, além da forte influência da administração e apoio a produção, que inserida no núcleo do sistema da empresa atua diretamente na gestão das operações de transformação.



FONTE: FRANSCISCHINI, 2007

FIGURA 2.2 – REPRESENTAÇÃO DA FUNÇÃO DA PRODUÇÃO SOB A ÓTICA DA ESTRATÉGIA

Com forte vínculo junto às operações estratégicas, a função produção desenvolve outros papéis além do mérito da produção dos bens e serviços. Slack et al (1997) os têm como: o apoio, a implementação e a impulsão da estratégia organizacional.

Tocante ao apoio estratégico, o autor considera que a produção tem o papel de desenvolver recursos que permitam que a organização atinja seus objetivos estratégicos. Isto compreende, por exemplo, a capacitação tecnológica de funcionários, superação de inovação tecnológica, desenvolvimento de estreita relação com fornecedores de peças, etc.

Outro papel da produção, segundo o mesmo autor, está na implementação da estratégia empresarial. É através da produção que, existindo alguma estratégia da empresa, será colocada em prática e poderá ser avaliada. Fica claro que é a administração da produção que faz todo o planejamento acontecer junto à função produção. O autor observa que mesmo na presença de uma estratégia original e brilhante, se a produção for inepta, poderá esta se tornar totalmente ineficaz.

O terceiro papel apresentado por Slack et al (1997) consiste na produção impulsionando a estratégia. Em outras palavras, significa ter a produção como líder, fornecendo meios para a obtenção de vantagem competitiva. Por exemplo, se o produto for bom, a entrega for rápida e o custo de produção for baixo, isto refletirá a certo prazo contribuindo para o sucesso da empresa no mercado; nota-se a dependência da produção na perpetuação da empresa no mercado.

Observada a influência direta dos SPs na estratégia posicional da empresa e definidas metas e estratégia, torna-se necessário formular planos para atingi-las, administrando os recursos humanos e físicos, direcionando a ação dos recursos humanos sobre os físicos e acompanhando esta ação, permitindo a correção de prováveis desvios. Neste contexto é que se promove a abordagem do Planejamento e Controle da Produção – PCP – como ferramenta de gerenciamento da produção.

## 2.2 PLANEJAMENTO E CONTROLE DA PRODUÇÃO (PCP)

Responsável pela coordenação e aplicação dos recursos produtivos de forma a atender, da melhor forma possível, aos planos estabelecidos, o PCP tem suas atividades exercidas nos três níveis hierárquicos de planejamento e controle das atividades produtivas de um sistema de produção: estratégico, tático e operacional. Respectivamente, cada nível define planos de longo, médio e curto prazo. A Figura 2.3 ilustra esta situação.



FONTE: ADAPTADO DE TUBINO, 2000 (PÁG. 25)

FIGURA 2.3 – VISÃO GERAL DAS ATIVIDADES DO PCP

No primeiro nível, o PCP participa da formulação do *Planejamento Estratégico da Produção*. Já no nível tático, o PCP desenvolve o *Planejamento-mestre da Produção*, de onde resultará o *Plano-mestre de Produção*. Por fim, no nível operacional, o PCP prepara a *Programação da Produção*, administrando estoques, seqüenciando, emitindo e liberando ordens de compras, fabricação e montagens, bem como a execução do *Acompanhamento e Controle da Produção* (TUBINO, 2000).

Para que fique melhor definida a atuação do PCP em cada um destes níveis hierárquicos, serão abordados de acordo com Tubino (2000), a seguir:

a) *Planejamento estratégico da produção*: estabelece um plano de produção amplo para um período de longo prazo, conforme estimativa de vendas e disponibilidade de recursos financeiros e produtivos, antevendo tipos e quantidades de produtos esperados para venda dentro do horizonte de planejamento. A capacidade de produção é fator físico limitante do processo produtivo, podendo ser incrementada ou reduzida, mediante planejamento prévio e adição de recursos financeiros;

b) *Planejamento-mestre da produção*: estabelece um Plano-Mestre de Produção - PMP - de produtos finais, detalhado em médio prazo, período a período, a partir do Plano de Produção. São definidos os itens que irão compor as famílias definidas no campo estratégico, assumindo compromissos de fabricação e montagem. Após, avalia-se os recursos produtivos, identificando gargalos que inviabilizem este plano em execução em curto prazo. O planejamento é iterativo, até que, tomadas todas as medidas corretivas sobre os PMP's, tenha-se o plano viável;

c) *Programação da produção*: Com base no PMP e nos registros de controle de estoques, a Programação da Produção estabelece em curto prazo, quanto e quando comprar, fabricar ou montar de cada item necessário à composição dos produtos finais. Em função da disponibilidade dos recursos produtivos, a Programação da Produção encarrega-se de fazer o sequenciamento das ordens emitidas, de forma a otimizar a utilização dos recursos. Se o Plano de Produção providenciou os recursos necessários e o PMP equacionou os gargalos, não deverá ocorrer problemas na execução do programa de produção seqüenciado. Assim, a Programação da Produção enviará as ordens a todos os setores responsáveis (empurrando) ou apenas à montagem final (puxando).

d) *Acompanhamento e controle da produção*: através da coleta e análise dos dados, visa garantir que o programa de produção emitido,

seja executado a contento. Quanto mais rápido os problemas forem identificados, mais efetivas serão as medidas corretivas objetivando o cumprimento do programa de produção.

Observa-se que o objeto de interesse deste trabalho está contido na administração da produção. É na esfera de curto-prazo do Planejamento e Controle da Produção (PCP), especificamente na programação da produção com o escalonamento das atividades. Pinedo e Chao (1999) consideram este escalonamento como fundamental na tomada de decisão, tanto de manufatura como de serviços. Walter (1999) acrescenta que organizar os processos produtivos frente a um planejamento maior é objetivar um melhor atendimento de prazos ou datas de entrega, minimização de tempos de fluxos dos estoques intermediários, maximização da capacidade disponível.

## 2.3 PROGRAMAÇÃO DA PRODUÇÃO E SEQUENCIAMENTO DE TAREFAS

Discutida a importância do PCP como método de gestão da produção, torna-se importante abordar o sequenciamento de tarefas na programação da produção como etapa fundamental e indissociável deste planejamento.

Sendo uma ferramenta de apoio a decisão importante (o que, quanto e quando produzir e comprar, além dos recursos necessários), o PCP sofre e influencia nas decisões de desempenho e eficiência. Como a programação da produção e o sequenciamento de operações são suas atividades de curto-prazo, herdaram esta característica (FERNANDES E GODINHO, 2007).

A programação fornece uma base para alocar tarefas aos centros de trabalho. É uma técnica de controle de capacidade que ressalta a sobrecarga e as ociosidades, além de especificar a ordem de execução das tarefas em cada centro de trabalho e o tempo em que cada trabalho começa e termina em cada máquina designada (HEIZER E RENDER, 2001 apud LANDMAN, 2005 e DOMINGOS et al, 2008).

Indicando a forte relação entre a classificação dos sistemas de manufatura e as atividades de sequenciamento, cita-se Taylor e Bolander (1994) apud Porter et al (1999) que conclui, ao organizar e comparar os atributos de produção por jobs (*job-shop*) e por fluxo (massa/contínua), existir diferenças significativas entre as escolhas apropriadas de planejamento e *scheduling*.

Tubino (2000), considerando a classificação por tipo de

operações, salienta que no seqüenciamento nos sistemas contínuos privilegia-se a velocidade do fluxo, contudo nos discretos, especificamente no dos repetitivos em massa, o seqüenciamento irá buscar o equilíbrio do ritmo entre os postos de trabalho. Nos repetitivos em lotes, o seqüenciamento visa priorizar as ordens através do uso de regras, que indicarão que tarefa irá ser despachada para operação em recurso produtivo, também definido. Por fim, em sistemas discretos, repetitivos por projeto, dada sua característica ímpar de demandar alto tempo de execução, o seqüenciamento tem foco no atendimento de prazos definidos, mediante o caminho crítico.

Dada a estrutura do PCP, o seqüenciamento e a emissão de um programa de produção deveriam ser tarefas simples, posto que existe também o suporte de um plano de produção de longo e de médio prazo. Contudo, instabilidades de curto prazo, como cancelamentos, adiamentos ou acréscimos de pedidos dos clientes, alterações nas especificações dos itens, deficiências na qualidade dos itens, etc., fazem com que a eficiência do sistema produtivo dependa fundamentalmente de um processo dinâmico de agendamento e emissão do programa de produção (TUBINO, 2000).

Único pela sua organização e especificidade de produtos que fabrica cada sistema produtivo pode ser classificado em função de alguns critérios, visando melhor compreendê-lo e, assim, tirar melhor proveito dos métodos de gestão da produção (ALMEIDA, 1995).

## 2.4 CLASSIFICAÇÕES DOS SISTEMAS PRODUTIVOS

Nas empresas, em sua incessante busca por melhores posicionamentos estratégicos, o autoconhecimento é um fator crucial para analisar as oportunidades e o potencial competitivo. Diversos autores têm proposto classificações dos sistemas de manufatura, dentre eles Almeida (1995), Porter et al (1999) e Perales (2001).

Relacionando as formas mais evidentes, os critérios de classificação têm sua importância no entendimento das características de cada sistema de produção e da sua relação com a complexidade das atividades de planejamento e controle dos sistemas. Assim, tem-se destacadas três categorias básicas:

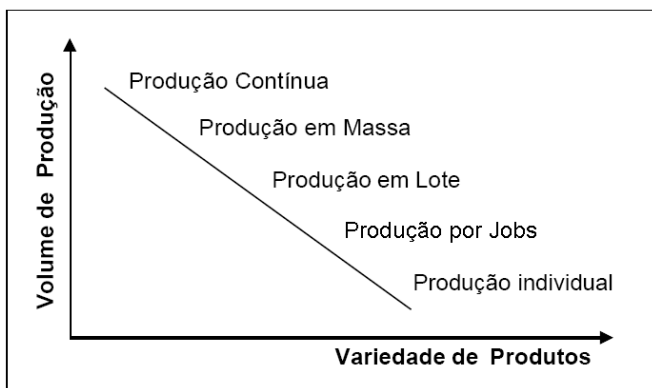
- *por tipo de operação*, distinguindo processos contínuos de discretos. Aos contínuos, associam-se os bens de base (energia elétrica, p. ex.), além da produção e demanda com alta uniformidade. Os discretos, ao contrário, podem ser isolados em lotes ou unidades e são sub-classificados em massa, lotes ou por projeto; Almeida (1995) toma a

produção contínua como *flow-shop* e a produção descontinuada como *job-shop*.

- *pela natureza do produto*, classificando através das definições de bens ou serviços, tendo-se os bens como produtos tangíveis e serviços como intangíveis;

- *pela resposta à demanda*, observando os critérios de orientação da produção, sejam estes voltados para estoque ou ordens/encomendas. Os primeiros operam com a venda dos produtos após sua produção. Nos seguintes, a venda ocorre antes da produção. A chave está no nível de interferência do cliente no produto final.

Sob a linha da Figura 2.4, que ilustra a forma de produção quanto ao volume de produção e variedade de produtos, Porter et al (1999) excursiona da produção em massa até a produção individual: da produção com alta padronização e volume até a outra, de alta flexibilidade na manufatura e necessidades específicas de clientes.



FONTE: ADAPTADA DE PORTER ET AL (1999)

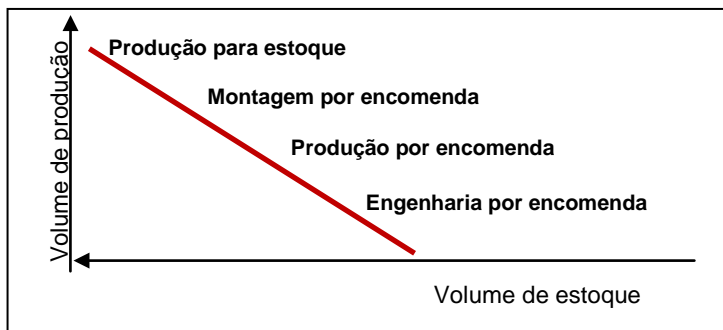
FIGURA 2.4 – RELAÇÃO VOLUME DE PRODUÇÃO X VARIEDADE DE PRODUTOS

Uma maior variedade de produtos imprime maior necessidade de flexibilidade na manufatura e, conseqüentemente, de qualificação da mão-de-obra. Além disso, o grau de interação dos clientes com o projeto e produção dá-se inversamente proporcional ao volume produzido, haja visto que grandes volumes consideram demandas estáveis e projetos com pouca ou nenhuma alteração em curto prazo (estruturas produtivas altamente especializadas e *layouts* por produto). No outro extremo, com clientes interagindo intimamente na produção, tem-se a manufatura

organizada por processo (ou células de manufatura), altamente flexível.

Matsuzaki (2004) reforça Porter et al (1999) ao tratar de flexibilidade e grau de automação, evidenciando a classe *job-shop* na produção de pequenos volumes e grande variedade de processos simultâneos, onde são fortes os investimentos em manufatura flexível. A alta flexibilidade e um efetivo polivalente e altamente qualificado atendem diferentes pedidos dos clientes e flutuações de demanda, esta última difícil de prever.

No tocante a resposta à demanda, Almeida (1995) relaciona que uma produção do tipo *flow-shop* está orientada a estoque, enquanto que a *job-shop* está orientada a encomenda. Já Porter et al (1999), Goulart (2000) e Perales (2001) apresentam alguns tipos básicos relacionando volume de produção e de estoque. Esta classificação guia o processo de gestão da produção de uma empresa, independente do seu setor de negócio. Na Figura 2.5 tem-se esta representação.



FONTE: ADAPTADA DE PORTER (1999); TRADUÇÃO PRÓPRIA  
FIGURA 2.5 – RELAÇÃO VOLUME DE PRODUÇÃO X VOLUME DE ESTOQUE

Sob a linha definida no gráfico, pode-se excursionar entre um processo produtivo totalmente voltado a disponibilidade de estoque (Produção orientada para estoque, *Make-to-stock* - MTS), com produção padronizada e baseada em previsão de demanda, até outro, com características adversas. Na engenharia sob encomenda (ou *Engineering to Order* - ETO), o projeto do produto é quase que totalmente orientado pelas especificações do cliente.

No MTS, o rápido prazo de entrega, fruto da pronta disponibilidade nos estoques contrasta com os custos gerados por este armazenamento, que tendem a ser grandes.



Classificados entre a montagem sob encomenda (*Assemble to order* - ATO) e a produção sob encomenda (*Make to order*- MTO), a primeira caracteriza-se na existência de subconjuntos, grandes componentes e materiais diversos que são armazenados até o recebimento dos pedidos dos clientes contendo as especificações dos produtos finais. A interação entre clientes e projeto de produto é limitada. As entregas dos produtos tendem a ser de médio a longo prazo e as incertezas nas demandas (*mix* e volume dos produtos) são gerenciadas pelo dimensionamento dos estoques de subconjuntos das áreas de montagem.

Na produção por encomenda (MTO), observa-se maior interação junto ao cliente, onde o projeto básico é desenvolvido a partir dos primeiros contatos com este, mas a produção tem início somente estando formalizado o pedido. Instabilidades como modificações durante o andamento da produção, são possíveis. A entrega varia de médio a longo prazo e as listas de materiais são quase sempre únicas para cada produto.

Avaliando o posicionamento estratégico de uma empresa e a conseqüente organização do seu processo produtivo, Araújo (2006) salienta que classes com maior interação do cliente são tendências atuais, haja visto que a esta interferência é fundamental na definição do modelo de programação e controle do sistema produtivo. Esta característica dinâmica requer modelos também dinâmicos de controle.

Domingos et al (2008) apontam que, além da importância dos agendamentos eficientes na programação da produção, é um tema discutido pela forte abrangência dos *schedules* (agendamentos) em diversas áreas da pesquisa operacional com objetivos de reserva de recursos no tempo, além do que aqueles sistemas classificados como do tipo *job-shop* são os que instauram maior complexidade de solução.

Discutidas as principais relações entre diferentes classes de sistemas produtivos e as contribuições de um seqüenciamento efetivo, ocorre ainda que diversas empresas apresentam sistemas mistos; combinações destes sistemas, como por exemplo: produzir em lotes na fabricação de peças e de forma contínua, nas linhas de montagem dos produtos acabados. Nestes casos, o PCP deverá operar também de forma híbrida (PORTER ET AL, 1999 e TUBINO, 2000).

## 2.5 JOB-SHOP COM PEQUENOS LOTES E PRODUÇÃO POR DEMANDA

O sistema de manufatura *job-shop* é a classe escolhida nesta tese para estudo e proposição de algoritmo onde objetiva-se agendamentos

eficientes de suas tarefas dentro de alguns parâmetros de qualidade e alicerçados nas características de produção onde se tem grande variedade de produtos e pouco volume de produção. Aliando-se às características de resposta a demanda e considerando-se a potencialidades da *lean production* (produção enxuta) é desejado não dispor de estoques, logo se prima por uma produção “puxada”, o que caracteriza produção por encomenda, seja para fornecimento a terceiros (intermediando etapa de produção), seja para o usuário final. Outras formas serão melhor discutidas posteriormente, mas fica previamente definida a classe de problema tratado.

De acordo com Lucero (2001), o usual para operar na classe *Job-shop*, nas últimas décadas, é fazendo resposta à demanda segundo a ordem do cliente (*make-to-order*). Isto é feito desde que ficou demonstrada a vantagem de trabalhar com o conceito de produzir *just-in-time*, com lotes de tamanho o menor possível. Outras características também definem a manufatura nestes sistemas, a saber:

a) pequenas quantidades por ordem de produção (quase unitárias);

b) arranjo funcional de chão de fábrica, podendo chegar a um arranjo celular por atividades, quando a maioria dos produtos é em lotes. Este arranjo seria intermediário entre o funcional e o celular por produtos (células virtuais, por ex.);

c) mudanças frequentes na engenharia de produtos;

d) os produtos são sempre projetados ou adaptados de projetos genéricos para os clientes dos *jobbing*, enquanto que para *batches* (lotes), são fabricados em pequenas quantidades;

e) os produtos ou peças que freqüentemente entram no sistema de manufatura mantêm memória de planos, dispositivos, ferramentais, etc. enquanto que nos produtos ou peças originais esta memória é desnecessária;

f) as ordens de produção são disparadas pelos métodos comprar materiais sob encomenda (*purchase-to-order*) / fazer contra pedido (*make-to-order*);

g) intensa entrada de produtos/componentes originais;

h) as listas de materiais (BOM) podem ser ora extensas e complexas, visando à sincronização para montagem, ou mesmo curtas e simples, apenas seriadas, procurando completar um produto através de uma série de operações;

i) as operações de fabricação podem ser simples ou complexas (p.ex: indústria de refrigerantes e fundição, respectivamente);

j) o produto tem de ser programado para conseguir cumprir os

prazos estipulados com o cliente no momento de se negociar a ordem de produção, pois eles interessam-se tanto em prazos de entrega quanto no preço do produto.

### 3. ESCALONAMENTO EM SISTEMAS DE PRODUÇÃO

Este capítulo apresenta aspectos relevantes ao tema de escalonamento de operações em ambientes de manufatura. Partindo da definição do problema de escalonamento em manufatura, em um segundo momento parte por evidenciar a classe *Job-shop* dentre os principais tipos de escalonamento existentes, cada qual com suas particularidades. Considerando que este trabalho motiva-se a utilizar técnicas heurísticas na busca por soluções eficientes junto aos *Job-shop scheduling problem* – JSSP, como são conhecidos na literatura serão apresentadas algumas das principais técnicas utilizadas neste intuito.

Posteriormente serão apresentadas algumas diferenças entre os casos clássicos e considerações sobre casos voltados à ambientes reais de produção.

#### 3.1 AGENDAMENTO DE TAREFAS

Em linhas gerais, o escalonamento consiste na alocação temporal de recursos para execução de determinadas operações, respeitando restrições impostas para cada caso. Não é difícil de relacioná-lo com diversos problemas que cercam constantemente a sociedade e que, mesmo passando despercebidos, são de fundamental importância. Cita-se, por exemplo, o fato de organizar a agenda pessoal da semana, como um escalonamento (*scheduling*), onde se prima por alocar o máximo de tarefas possível a um recurso (respeitando suas sequências lógicas de operação e seus tempos de conclusão) de maneira a aproveitar melhor o tempo e concluí-las todas, com êxito, com eficiência. Outros exemplos, como a alocação de professores às disciplinas, considerando restrições de salas, pré-requisitos de alunos e especialidade do profissional, ou ainda em hospitais, na organização do trabalho mediante alocação de médicos ou leitos à pacientes, são problemas igualmente relacionados.

Na indústria, tendo máquinas, tarefas a realizar, tempos de produção, prazos de entrega, etc., os problemas de agendamento de tarefas surgem como uma necessidade de organização de trabalho na busca de eficiência. Considerando a disputa de mercado entre empresas de uma mesma indústria, além das dimensões globais assumidas por estes ambientes competitivos, obter uma vantagem competitiva é fator

crucial na obtenção do sucesso.

Os agendamentos de operações no âmbito industrial têm sido extensamente estudados desde meados de 1950, mas tem ganhado mais força com o proeminente avanço das ferramentas computacionais, que permitem a implementação de algoritmos mais elaborados, muitas vezes explorando combinações de técnicas já bastante eficientes na solução de problemas reais. Isto se deve, posto que, conforme Lenstra et al (1977) e atualmente Araujo (2006), os problemas de escalonamento são da classe de problemas combinatórios NP - difíceis, ou seja, exigem a enumeração explícita ou implícita de todas as alternativas possíveis para garantir a solução ótima. Assim, observa-se que apesar de existirem métodos exatos, é quase impossível resolvê-los desta forma, exceto para exemplos relativamente pequenos do problema. Carlier e Pinzon (1989) apresentam uma forma baseada em ramificação e corte (*branch and bound*), que permitiu solucionar o problema FT10 (10x10) proposto por Fisher e Thomson (1963), até então não resolvido, mesmo empregando técnicas diferentes. Contudo, tais aplicações são de relevância apenas teórica devido a sua complexidade exponencial de tempo de execução (BRAUNE ET AL, 2004).

Em ambientes de produção reais, a provável otimalidade não é o critério que deva ser satisfeito. É suficiente obter resultados próximos do ótimo, mas em um tempo razoável, conseguido com o uso de métodos heurísticos. São aproximações utilizadas na solução dos JSSP: Busca Tabu (TS) (BARNES e CHAMBERS, 1995), *Simulated Annealing* (SA), algoritmos genéticos (AG) (YAMADA E NAKANO, 1997) e *Particle Swarm Optimization* – PSO (ZHAO, ZHANG e YANG, 2006).

Tocante ainda aos casos reais, diversos autores têm chamado a atenção para as características dinâmicas do problema de agendamento de atividades na indústria. Cita-se aqui os trabalhos de Araújo (2006), Bierwirth e Mattfeld (2004), Candido (1997), dentre tantos outros, com a aplicação de algoritmos genéticos em JSSP baseados em sistemas de manufatura focados na produção *make-to-order* (MTO), já apresentados como fortemente customizada pelo cliente, e, portanto com características bastante dinâmicas.

A fim de melhor especificar os problemas de alocação de recursos e suas restrições, parte-se por classificá-los mediante notação proposta por BRUKER e utilizada por Araújo (2006) e Goldman (2007). Intrinsecamente será abordada a característica estática e dinâmica dos *job-shops*, como também as regras que definem e diferenciam casos clássicos de realistas.

### 3.2 CLASSES DO PROBLEMA DE AGENDAMENTO DE TAREFAS

A organização dos recursos produtivos para obter maior eficiência na manufatura é um problema difícil. Assim menciona Herrmann et al (1995) quando aponta que a produção de peças com mesmas rotas e tempos de *setup* curtos pode obter resultados eficientes com a aplicação da filosofia *just-in-time* (JIT) em sistemas puxados. Em muitos processos de manufatura, entretanto, encontrar uma solução ótima é muito mais complicado (por exemplo, pela grande variedade de rotas e/ou combinações entre recursos e peças a designar).

Extensamente abordados na literatura, os problemas de escalonamento de tarefas envolvendo  $m$  máquinas e  $n$  tarefas tem sido solucionados por técnicas exatas e heurísticas. Yamada e Nakano (1995) definem um agendamento como sendo a ordenação entre todas as operações que devem ser processadas na mesma máquina, considerando as restrições de precedência. Tais problemas são também mencionados por Taillard (1992), ao propor comparações entre os tipos clássicos de *scheduling* encontrados: *flow-shop*, *job-shop* e *open-shop*. As particularidades de cada um são melhor evidenciadas a seguir.

#### 3.2.1 Notação de Bruker

Nesta seção apresenta-se a notação introduzida por Bruker (1998), que classifica diversos tipos de escalonamentos, sendo composta pelos campos:  $\alpha|\beta|\gamma$ , referentes a recursos, tarefas e critério de otimização, respectivamente. Sendo referenciada em diversos trabalhos e atualmente também por Araujo (2006) e Goldman (2007), ressalta-se que a abordagem deste tópico nesta tese tem o papel de auxiliar na compreensão destes problemas, uma vez que evidencia e define diversos elementos envolvidos no tema, sendo elemento ímpar na definição da classe dos problemas de *Job-shop*, objetivo deste trabalho. A seguir serão evidenciados os campos envolvidos na notação supramencionada.

##### 3.2.1.1 Campo $\alpha$ - Recursos

Basicamente, como aponta Araujo (2006), os tipos mais comuns de recursos em sistemas de manufatura são as máquinas paralelas e as do tipo “*shop*” (especializadas). No caso das primeiras, as tarefas podem ser completadas por uma única máquina. Para o caso das especializadas, existem tarefas que podem precisar ser executadas em máquinas distintas. As máquinas paralelas não pertencem ao escopo desta tese.

### 3.2.1.1.1 Máquinas *Shop* (Especializadas)

Considerando-se que o foco desta tese está sobre os sistemas de manufatura composto por máquinas especializadas, como também salienta Taillard (1992), três são os modelos de organização destas máquinas, caracterizando os sistemas conforme Tabela 3.1:

TIPO	DESCRIÇÃO
Flow-shop ( $F_m$ )	No modelo <i>flow-shop</i> ( <i>line-transfer</i> ), todas as tarefas têm o mesmo número de operações as quais são executadas na mesma ordem, por todas as $m$ máquinas em série. Após a execução em uma máquina, a tarefa entra na fila para a próxima máquina. Geralmente as filas seguem a disciplina FIFO ( <i>first in - first out</i> );
Open-shop ( $O_m$ )	Neste modelo as tarefas também devem ser executadas por todas as $m$ máquinas, entretanto, alguns tempos de execução podem ser nulos. Não existem restrições quanto à ordem das máquinas onde cada uma das tarefas deve passar;
Job-shop ( $J_m$ )	Neste modelo cada uma das tarefas possui um roteiro pré-determinado, O número de máquinas, como nos modelos anteriores também é $m$ . Um tipo especial de job-shop onde uma tarefa pode visitar uma máquina mais do que uma vez, é denominado por recirculação (com a palavra <i>recrc</i> no segundo campo).

FONTE: ADAPTADO DE BRUKER (1998) APUD ARAUJO (2006)

TABELA 3.1 - VARIAÇÕES DE MÁQUINAS DO TIPO *SHOP* (ESPECIALIZADAS).

### 3.2.1.2 Campo $\beta$ - Tarefas

Aqui são introduzidas algumas características bastante comuns relativas às tarefas a serem escalonadas e relacionadas na Tabela 3.2.

No caso de problemas de escalonamento que não se adaptem as variações apresentadas na tabela, pode-se definir novos parâmetros para os campos, desde que a notação de Brucker para os campos recurso, tarefa e critério de otimização sejam respeitados (BRUCKER, 1998).

TIPO	DESCRIÇÃO
Tempo de preparação ( <i>setup</i> ) da máquina ( $S_{jk}$ )	Dadas duas tarefas $J_j$ e $J_k$ , $S_{jk}$ corresponde ao tempo de preparação de uma máquina que executou uma tarefa $J_j$ e vai executar uma tarefa $J_k$ . Caso este tempo também dependa da máquina $m_i$ em questão, usa-se a notação $s_{ijk}$ . No caso de ser a primeira (ou última) tarefa executada na máquina, o tempo de preparação é denotado por $s_{0j}$ e $s_{j0}$ , respectivamente.
Relações de precedência ( <i>prec</i> )	Quando existir, a ordem para a execução das tarefas geralmente é representada por um grafo orientado, onde os vértices correspondem às tarefas e os arcos às relações diretas de precedência. Um arco de uma tarefa $J_j$ a uma tarefa $J_k$ faz com que a tarefa $J_k$ possa iniciar sua execução apenas após o término da tarefa $J_j$ . Caso o grafo de precedência pertença a uma família específica, ao invés de <i>prec</i> , coloca-se o nome da família no segundo campo. Os mais comuns são: <i>intree</i> , <i>outree</i> , <i>chains</i> , <i>split-computemerge</i> , dentre outros.
Tempo de comunicação entre máquinas ( $Tc_{jk}$ )	No caso de precedência entre duas tarefas $J_j$ e $J_k$ , se a tarefa $J_k$ finaliza a sua execução no instante $t$ na máquina $m_i$ , a tarefa $J_k$ não pode iniciar a sua execução antes do instante $t$ , na mesma máquina $m_i$ , ou do instante $t+c_{jk}$ em qualquer outra máquina.
Interrupção ou preempção ( <i>prmp</i> )	Quando é possível interromper uma tarefa, ela não precisa necessariamente terminar na mesma máquina em que começou a sua execução. O tempo de execução utilizado até a interrupção não é perdido. Quando a tarefa volta a uma máquina qualquer, ela só precisa ser executada durante o tempo restante após a última interrupção.
Restrições de escolha de máquina ( $M_j$ )	Este item aparece no segundo campo quando a tarefa $J_j$ só puder ser executada em um sub-conjunto $M_j \subset P$ .

FONTE: ADAPTADO DE BRUKER (1998) APUD ARAUJO (2006)

TABELA 3.2 - ALGUMAS CARACTERÍSTICAS COMUMENTE ASSOCIADAS AO CAMPO DE TAREFAS.

### 3.2.1.3 Campo $\gamma$ - Critério de otimização

De acordo com Araujo (2006) citando Bruker (1988), para a definição dos principais critérios de otimização usados em problemas de escalonamento, é importante conhecer o instante em que cada tarefa  $J_j$  termina a sua execução, denotado por  $Tc_j$ . Associa-se então, uma função de custo  $f_j(Tc_j)$  a cada tarefa, onde o custo total para a maioria dos critérios de otimização corresponde a:

$$\max \left\{ f_j(Tc_j) \mid 1 \leq j \leq n \right\} \quad (3.1)$$

ou a soma de todos os custos:

$$\sum_{j=1}^n f_j(Tc_j) \quad (3.2)$$

Tem-se por  $d_j$  a data devida do job<sub>i</sub>.

Como os problemas de escalonamento consistem em minimizar a função de custo total, na Tabela 3.3 Bruker (1988) relaciona as funções de custo mais comuns:

TIPO	DESCRIÇÃO
<i>Makespan</i> ( $Tc_{\max}$ )	Definido como o instante de tempo em que a última tarefa é finalizada, $Tc_{\max} = \max \{Tc_1, \dots, Tc_n\}$ . Neste caso o critério corresponde ao máximo e as $f_j()$ funções são funções identidade.
Tempo total de término $(\sum C_j)$	Definido como a soma dos instantes de término de cada tarefa, ou seja, $\sum Tc_j = \sum_{j=1}^n Tc_j$
Tempo total de término ponderado $\sum w_j \cdot Tc_j$	Definido como soma dos instantes de término de cada tarefa com peso $w_j$ , ou seja, $\sum w_j \cdot Tc_j = \sum_{j=1}^n w_j \cdot Tc_j$
Grau de atraso (lateness)	$L_j = Tc_j - d_j$ , podendo assumir valores negativos
Grau de adiantamento (earliness)	$E_j = \max \{0, d_j - Tc_j\}$
Grau de atraso efetivo (tardiness)	$T_j = \max \{0, Tc_j - d_j\}$
Penalidade unitária por atraso	$U_j = \begin{cases} 0, & \text{se } Tc_j \leq d_j; \\ 1, & \text{se } Tc_j > d_j; \end{cases}$

FONTE: ADAPTADO DE BRUKER (1998) APUD ARAUJO (2006)

TABELA 3.3 – FUNÇÕES DE CUSTO MAIS COMUNS.

Da mesma forma que o *makespan*, definido por  $\max \{Tc_1, \dots, Tc_n\}$ , é denotado por  $Tc_{\max}$ , os demais critérios de maximização o são por  $L_{\max}$ ,  $E_{\max}$ ,  $T_{\max}$  e  $U_{\max}$ .

Fazendo-se uso da notação de Bruker, pode-se descrever o



problema de JSSP tratado nesta tese como o descrito na E.q. (3.3), onde considerando os tempos de execução  $P_{ij}$ , relações de precedência arbitradas por grafos de precedência em  $m$  máquinas especializadas do tipo *job shop*, de forma que os atrasos sejam minimizados, onde o critério de otimização segue a utilização de heurísticas híbridas resultante da combinação de técnicas evolutivas e regras de despacho combinadas:

$$J_m \mid prec; P_{ij}; Tci_j; t; d_i; Si_{jk}; M_j \subset P \mid \text{Regras de Despacho} \quad (3.3)$$

Neste ponto, já tendo sido realizada uma abordagem para elucidar os principais problemas de escalonamento mediante notação de Bruker, torna-se importante abordar a definição apresentada por Yamada e Nakano (1997) aos JSSP. Tais problemas apresentam-se como um grupo de  $n$  tarefas  $\{J_j\}_{1 \leq j \leq n}$  a serem processadas em um conjunto de  $m$  máquinas  $\{M_v\}_{1 \leq v \leq m}$ . Cada tarefa tem uma seqüência tecnológica de máquinas á ser processadas. A execução da tarefa  $J_j$  na máquina  $M_v$  consiste na operação  $O_{jv}$ . Tal operação requer a exclusividade de uso da máquina  $M_v$  por um tempo de processamento necessariamente ininterrupto de duração  $P_{jv}$ . É, portanto, um problema com um grupo de tempos a completar para cada operação  $\{O_{jv}\}_{1 \leq j \leq n, 1 \leq v \leq m}$  que satisfaça as restrições do problema  $C = \{C_1, C_2, \dots, C_c\}$ . O tempo para completar todas estas tarefas é o de *makespan*  $L$ . Minimizá-lo, é objetivo tradicional dos JSSP clássicos (VÀZQUEZ e WHITLEY, 2000).

### 3.2.2. Casos de JSSP clássicos e realistas

Algumas regras são adotadas para definir os JSSP clássicos (teóricos), como sugere Carlier e Pinzon (1989) além de French (1982) :

- a) todas as máquinas estão disponíveis no tempo zero. Respeitadas as relações de precedência, todas as operações podem ser executadas a partir do início do processo de escalonamento;
- b) cada máquina pode processar apenas uma operação por vez;
- c) existe apenas uma unidade de cada tipo de máquina;
- d) as máquinas podem ficar ociosas;
- e) uma máquina é o único recurso requerido para processar uma operação;
- f) o *buffer* para o processo é ilimitado;
- g) o sistema não é preemptivo: uma vez iniciada, a operação deverá ser completada antes que outra seja iniciada, na mesma máquina;

h) cada tarefa tem o mesmo número de operações, sendo uma operação por máquina. Não é permitido que duas ou mais operações de uma mesma tarefa sejam executadas em uma mesma máquina;

i) o sistema é estático, ou seja, novas tarefas não podem ser inseridas no *scheduling* durante sua execução e também, não pode ocorrer falhas ou quebras de recursos (máquinas);

j) todos os dados do problema são determinísticos, ou seja, é sabido o número de tarefas, máquinas e tempos de processamento;

k) tempos de transporte e *setup* para cada máquina são desconsiderados;

Observa-se que muitas destas restrições não são compatíveis com a realidade das empresas, entretanto encontra-se na literatura uma variedade de trabalhos na busca de solução destes problemas teóricos, seguindo tais regras de execução. Os seus principais objetivos versam, dentre outros, sobre formas de garantir melhores resultados em métodos heurísticos, ou comparativos de técnicas empregadas. Muitas vezes, na expectativa de aproximar casos teóricos de casos reais, necessita-se adicionar ou mesmo relaxar estas restrições e tanto métodos heurísticos diversos como *benchmarks* aplicados e estes casos também são muito pesquisados na atualidade.

No tocante a casos realistas, o ambiente produtivo é que irá determinar a dimensão e tipo de restrições do JSSP. Assim, como menciona Candido (1997), tornar-se-ia impraticável desenvolver um sistema de *scheduling* universal, dadas as características intrínsecas de cada ambiente produtivo. Entretanto, algumas delas repetem-se com certa frequência, e assim é possível propor sistemas de agendamento que contemplem tais casos, solucionando um conjunto grande de problemas industriais relacionados aos JSSP na prática.

### 3.2.3. Casos reais: características e estratégias envolvidas

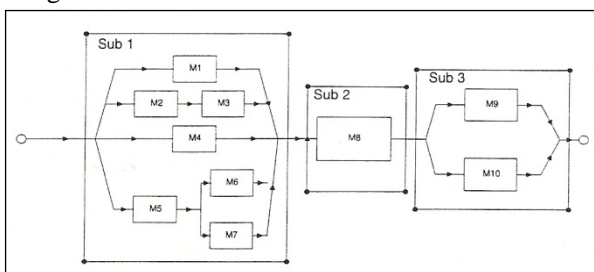
A seguir, têm-se algumas das características comumente encontradas em ambientes reais, agora detalhadas:

a) Restrições de precedência e capacidade dos recursos: restrições clássicas dos JSSP. Elas definem as relações de precedência entre operações de uma mesma peça a serem seguidas e que cada recurso processa, no máximo, uma operação por vez;

b) Requisição de recursos adicionais renováveis para execução de uma operação (ferramentas, pessoal, etc.): o processamento de uma operação pode requerer tipos variados de recursos. Diversas unidades de

cada tipo de recurso envolvido podem ser também requeridas. Contudo, apenas uma única máquina pode ser usada para processar uma operação.

c) Plantas alternativas de processamento para cada peça: uma peça pode ser processada por seqüências de máquinas diferentes. O autor, visando facilitar o modelamento de plantas alternativas de processamento, divide os processos de peças em seqüências de sub-processos. Cada sub-processo pode ser executado por rotas alternativas de operações, sendo que cada qual é formada por seqüências de máquinas. Araujo (2006) também salienta os benefícios ao desempenho do sistema quando há diversidade nas rotas de processo. O número de rotas alternativas é dado pelo número de rotas possíveis menos um, ilustrado a seguir:



FONTE: CANDIDO (1997)

FIGURA 3.1 – PLANO DE PROCESSO COM ROTAS ALTERNATIVAS NOS SUBPROCESSOS

No diagrama da figura 3.1 o processo da peça é dividido em três partes, ou sub-processos: Sub1, Sub2 e Sub3. Nota-se neste exemplo que Candido (1997) considera que um dado sub-processo pode ser realizado por um número diferente de máquinas alternativas. Esta idéia não é abraçada por esta tese, contudo fica ilustrada a possibilidade de ocorrência noutras instâncias. Quanto ao número de rotas e rotas alternativas, observa-se para o Sub1 a existência de 5 rotas, sendo 4 alternativas. De forma análoga, para Sub 2 tem-se 1 e 0 respectivamente. Para Sub 3 tem-se 2 e 1, da mesma forma. No exemplo, é possível executar a peça pela seguinte seqüência de máquinas: M5, M7, M8, M9. Observa-se que, de Sub 1, tomou-se M5 e M7. como no Sub 2 tem-se apenas uma opção, toma-se M8, e considerando o Sub 3, optou-se por M9 como sendo a rota selecionada.

Os demais recursos (ex: pessoal, ferramentas) necessários ao processo da operação não são considerados desta mesma forma, por variar em tipo e quantidade. Em outras palavras, Candido (1997)

desconsidera rotas alternativas de sub-processos para recursos que não máquinas, atuando de forma relaxada sobre estes.

d) Recursos alternativos para cada operação: uma vez que podem existir mais de uma unidade de cada tipo de recurso na planta produtiva, podem existir também conjuntos alternativos de recursos para processar uma operação;

e) Calendários de máquinas, manutenção e pessoal (ex: manutenção preventiva, treinamento de pessoal, etc.): as máquinas tem suas paradas agendadas para manutenção e limpeza, ferramentas necessitam serem afiadas/restauradas, funcionários necessitam de treinamento, folgas e férias. Os recursos nos ambientes produtivos não são operacionais em 100% do tempo, logo, o calendário de cada um destes recursos deve ser levado em consideração;

f) tempo disponível para matéria-prima e produção externa de peças, representadas por tempos disponíveis para tarefas e operações: muito frequentemente, o processamento de uma operação requer matéria-prima ou componentes não produzidos na planta, mas adquiridos de fornecedores. Estes materiais e componentes estão disponíveis apenas após certa hora específica. O sistema de controle de inventário deve promover a disponibilidade de datas destes itens ao sistema de agendamento, que irá interpretá-las como operações e tempos disponíveis, uma vez que uma tarefa ou operação não pode ser iniciada até que toda matéria-prima e componentes necessários estejam disponíveis;

g) tempos de *setup* dependentes de seqüência: uma operação apresenta um tempo de *setup* dependente de uma seqüência, quando este tempo depende das operações as quais foram previamente processadas na máquina. Para tanto, uma tabela de *setup* deverá existir, para conformar as necessidades de cada uma destas operações;

h) sobreposição de lotes: por exemplo, seja um tamanho de lote igual a 40 e que a manufatura deste mesmo lote necessite de 2 operações. Sob certas condições, a segunda operação pode ser iniciada após apenas uma fração das 40 peças ter sido produzida pela primeira operação. Toma-se que, para cada 10 peças produzidas na primeira operação, elas são transportadas e a segunda operação é processadas pela (célula de) máquina seguinte. Apesar da sobreposição de lotes ser permitida, o processamento corrente da mesma operação não o é. Isto significa que uma operação pode iniciar sem que todas as peças da operação anterior tenham sido completadas, mas apenas um conjunto de recursos (com apenas uma máquina) é escolhido para processar uma operação. Lembra-se que preempção de lotes não é permitido, ou seja,

uma vez que a execução de uma operação é iniciada, ela não deve ser interrompida até que seja concluída, finalizando a produção do lote. A sobreposição da última operação de uma tarefa pode ser utilizada em modelamentos de envio (transporte/despacho de carga) parciais.

i) Função de avaliação multiobjetivo regular: as funções multiobjetivo regulares consideram somente combinações de medições regulares de performance, devido a movimentos no algoritmo de melhora local ser baseado em manipulação de cadeia ativa e o operador de recombinação do algoritmo genético estar embutido no gerador destes agendamento ativos. Cabe ao usuário definir a função multiobjetivo a aplicar. Uma avaliação de performance regular não pode ser melhorada pelo atraso no tempo de finalização de uma tarefa. A exemplo seguem alguns dos métodos de avaliação regulares mais utilizados, agrupados em conjuntos básicos: tempo médio de fluxo, tempo médio de finalização, tempo médio de espera, media de atraso; tempo máximo de finalização (*makespan*), tempo inativo médio de máquina, tempo total inativo de máquina, média do número de tarefas sendo processadas por unidade de tempo; máximo atraso; máximo tempo de fluxo; atraso médio. Apenas combinações lineares são consideradas no modelo do autor.

Candido (1997) menciona que um sistema que intenciona abranger uma larga faixa de JSSP reais deverá considerar restrições como as supracitadas e estar igualmente apto a operar com funções de desempenho multicritério, sob condições dinâmicas de trabalho.

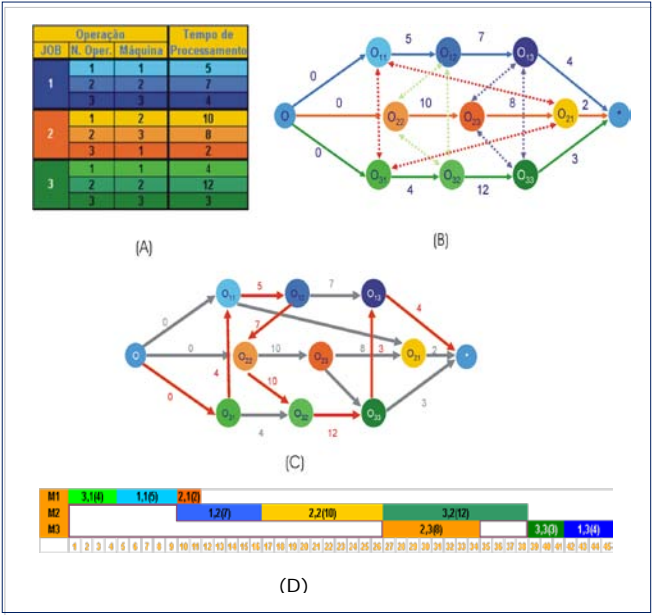
No caso de problemas que vislumbrem operar sobre pequenos lotes, caso deste trabalho, situações como *batch overlap* e quebras de lotes, podem ser desconsideradas, Quanto aos tempos de transporte e de *setup*, Lucero (2001) sugere operar com tabelas independentes onde são colhidos os respectivos valores conforme a sequência de operação, que posteriormente somados, irão definir o tempo de preparação antes do processamento da operação. Como o problema de pequenos lotes é bastante próximo do proposto por Candido (1997), muitas destas restrições serão seguidas, em detrimento de outras poucas, como as citadas a pouco, que poderão ser relaxadas.

### 3.3 AS FORMAS DE REPRESENTAÇÃO DOS JSSP

Um *scheduling* comumente é visto sob a forma de grafo, onde os seus nós são as operações com atributo igual ao tempo de processamento ( $P_{ij}$ ). Acrescentam-se os nós partida e chegada, com

$P_{ij}=0$ . As restrições de precedência são arcos direcionais, apontando da operação atual para sua sucessora. As restrições de máquinas são arcos bidirecionais (disjuntivos). Selecionando uma de suas duas direções de forma que não ocorram ciclos no grafo, reduz-se, destas para restrições de precedência. Estando o grafo totalmente orientado, encontrar o *makespan* é calcular o maior caminho entre a partida e a chegada. Assim, o JSSP reduz-se a encontrar, mediante trocas de orientações dos arcos no caminho mais longo, uma onde o *makespan* é mínimo (AYDIN E FOGARTY, 2004).

Na Figura 3.2 se observa a formação do grafo mencionado a partir de uma tabela de tarefas e operações a serem agendadas em certo grupo de máquinas. Conhecidas as restrições do JSSP, é possível encontrar soluções viáveis mediante a aplicação do caminho crítico sobre o grafo, após a orientação dos arcos disjuntivos.



FONTE: ADAPTADO DE CANDIDO (1997)  
FIGURA 3.2 –JSSP: (A) GRAFO DISJUNTIVO; (B) *MAKESPAN* (C, D).

Na Tabela 3.1 tem-se um exemplo de solução para um JSSP de dimensão 3x3. Os dados indicam as rotas para cada tarefa através de cada uma das máquinas, além dos tempos de processamento de cada operação (na Tabela 3.1, entre parênteses).

Tarefa	Rotas de operações (tempo de processamento)		
1	1(3)	2(3)	3(3)
2	1(2)	3(3)	2(4)
3	2(3)	1(2)	3(1)

FONTE: YAMADA E NAKANO (1997)  
TABELA 3.4 – PROBLEMA JSSP 3X3

Yamada e Nakano (1997) fazem uso também do gráfico de Gantt como forma conveniente de representar uma solução para os JSSP. Na figura 3.3(a) pode-se observar a representação da solução apresentada na Tabela 3.1, sob gráfico de Gantt.

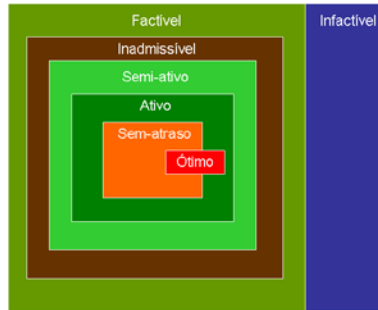


FONTE: YAMADA E NAKANO (1997)  
FIGURA 3.3 – SOLUÇÃO DE JSSP 3X3 – (A) GRÁFICO DE GANTT E (B) MATRIZ SEQUÊNCIA DE TAREFAS

Os JSSP também podem ser vistos sob a forma de problemas de ordenação, como os *Travelling Salesman Problem* (TSP). Em linhas gerais, um agendamento pode ser representado pela permutação de tarefas em cada máquina, ou seja, permutações *m*-particionadas com dimensões equivalentes as operações a serem realizadas, denominada *matriz de seqüência de tarefas* e ilustrada na Figura 3.2(b), considerando-se a Tabela 3.1.

3.4. AS CLASSES DE AGENDAMENTO PARA O JSSP

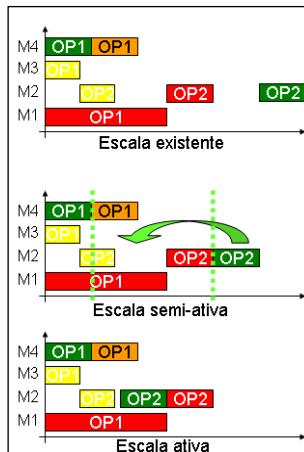
No tocante a factibilidade, existem algumas classificações para os agendamentos que são: inadmissíveis, semi-ativos, ativos e sem-atraso. Partindo-se para uma definição simplificada de cada tipo de classificação, tem-se que nos *schedulings* inadmissíveis, ocorrem excessos de tempos de inatividade que podem ser reduzidos pelo avanço de operações. Já os semi-ativos não apresentam tempos inativos, mas podem ser melhorados com ao avanço de operações sem atraso de outras. A Figura 3.4 os relaciona devidamente.



FONTE: ADAPTADO DE VÁSQUES E WITLEY, 2000

FIGURA 3.4 – RELAÇÃO ENTRE OS TIPOS DE SCHEDULES NO JSSP

Na Figura 3.5 se ilustra a diferença dos semi-ativos para os ativos. Estes últimos são agendamentos ótimos, pois não apresentam tempos de inatividade e nenhuma das operações poderá ser finalizada antes que outra operação seja atrasada.



FONTE: ADAPTADO DE BECK, 2000

FIGURA 3.5 – EXEMPLO ENTRE ESCALAS SEMI-ATIVAS E ATIVAS

A Figura 3.5 ilustra ainda os agendamentos sem-atraso, contidos no conjunto dos ativos, mas objetivando minimizar os tempos inativos das máquinas. Nenhuma máquina fica inativa se uma operação pode ser processada. A título ilustrativo, os algoritmos baseados no GT, que permitem construir escalas ativas e ativas sem-espera, observados na seção 3.7.2.2.



## 4. TÉCNICAS APLICADAS NA SOLUÇÃO DOS JSSP

Na literatura relacionada aos problemas de JSSP é possível observar uma grande diversidade de métodos utilizados na busca por soluções. Extensamente abordado, Jain e Meeran (1998) citam o trabalho de Johnson (1954) como um dos primeiros trabalhos expressivos na teoria de *scheduling*, que em linhas gerais objetivava minimizar o *makespan* para um sistema composto de duas máquinas. Diversos outros trabalhos o seguiram, onde a variedade de técnicas envolvidas e as formas de modelamento do problema tomam dimensões elevadas no decorrer destas quase seis décadas. Certas vezes, métodos diferentes mostram-se melhores que outros para mesmos problemas, mas esta diversidade é que enriquece o tema tratado. Salienta-se também a potencialidade crescente dos sistemas processados, que a cada tempo possibilita mais utilizar metodologias antes impraticáveis pelo custo computacional.

A seguir serão apresentados alguns métodos aplicados na solução dos JSSP, e como esta tese não tem a pretensão de extinguir a discussão sobre o tema, irá aportar apenas as mais expressivas técnicas utilizadas, bastante evidenciadas na literatura e tocante a este estado-da-arte. JAIN E MEERAN (1998) formulam um estudo que relaciona diversas destas técnicas, onde, para estudos comparativos mais profundos, merecem consulta por parte dos mais interessados. Neste trabalho, partindo das soluções exatas originárias da programação matemática e passando pelas soluções propostas pelos métodos aproximativos, intenciona-se evidenciar o tema e salientar pontos que serão importantes para a fundamentação desta tese.

### 4.1. TÉCNICAS EXATAS DE SOLUÇÃO

Dada a característica combinatória dos problemas de agendamento de tarefas, já apresentada anteriormente, apenas nos casos onde as dimensões dos problemas são pequenas é possível empregar métodos da pesquisa operacional visando buscar resultados ótimos.

Diversas estratégias são apresentadas pela literatura na solução dos JSSP clássicos. Em problemas de decisão deste tipo, o *Critical Path Method* – CPM é um dos bastante citados. Também formulações envolvendo Programação Linear (PL), Programação Linear Inteira (PLI) ou ainda Programação Inteira Mista (*Mixed Integer Programming* - MIP) são utilizadas. Também os métodos enumerativos, como o *Branch and Bound*, são fortes destaques e a programação dinâmica evidencia-se

na solução ótima dos JSSP clássicos.

Durante os anos 1950-60, no transcorrer da corrida armamentista entre EUA e a ex-URSS, a marinha americana realizou um programa de construção de submarinos nucleares e a força aérea, de aviões bombardeiros estratégicos de longo alcance. No ano de 1958 é desenvolvido pela empresa de consultoria Booz, Allen & Hamilton, ligada ao projeto da marinha americana, o método PERT - *Program Evaluation and Review Technique*, como instrumento de planejamento e controle, que contribuiu para o sucesso do projeto. Na mesma época, por demanda da empresa *Lockheed Aircraft Corporation*, empresa ligada ao projeto dos bombardeiros estratégicos e com o programa aeroespacial da NASA, as empresas Dupont e Univac desenvolveram o método CPM - *Critical Path Method* visando o cumprimento dos contratos firmados com o governo americano.

Assim, o CPM é um procedimento para análise de redes que identifica quais tarefas compõem o *caminho crítico*, ou seja, seqüência de tarefas onde qualquer atraso na finalização irá prorrogar também o tempo total do projeto. O CPM identifica formalmente as tarefas que devem ser finalizadas em tempo para que todo projeto possa ser concluído no prazo desejado, além de também identificar as tarefas que podem ser atrasadas se houver necessidade de realocação de recursos para absorção de tarefas perdidas. WALL (1996) aponta que o CPM é aplicado em problemas de agendamento sem restrição de recursos e com restrições de precedência, logo não pode resolver problemas com restrições no número de recursos disponíveis, presente na realidade dos ambientes industriais.

O PERT/CPM, junção dos métodos supracitados, deu-se devido a sua forte semelhança, empregando conceitos de rede (grafos) para o planejamento e avaliação da coordenação das atividades do projeto.

No tocante às formulações matemáticas, a intratabilidade intrínseca do problema é apontada por Giffler e Thompson (1960), que consideram não serem métodos práticos de solução. Em linhas gerais, são requeridas muitas simplificações aos problemas a fim de encontrar soluções, e ainda que se mostram pouco adaptáveis às variações de tamanho, ficando as aplicações, restritas a poucos problemas de pequeno porte (WALL, 1996). Não fica difícil imaginar que, para problemas reais envolvendo os JSSP, mais complexos que os clássicos, agravam ainda mais a aplicação destas estratégias.

Os estudos envolvendo JSSP clássicos e busca por soluções ótimas na minimização do *makespan* levaram diversos autores a empregar técnicas enumerativas. Nestas estratégias de solução, como

seu nome sugere, são enumeradas as possibilidades de solução e posteriormente é realizada a eliminação das soluções não-ótimas. Duas são as formas de enumeração encontradas: explícitas e implícitas. No primeiro caso, todas as soluções possíveis são listadas para então efetuar-se os cancelamentos. Já no segundo caso, utiliza-se em geral heurísticas para limitar o tamanho das árvores geradas, possibilitando assim resolver problemas de dimensões um pouco maiores, entretanto ainda limitados, se comparados a casos reais (WALL, 1996). Ainda WALL (1996) citando Sprecher e Drexl (1996), considera que métodos enumerativos não podem resolver grandes problemas, haja visto que a árvore torna-se grande. Apesar disso, bastante citada na literatura, à estratégia enumerativa *Branch and Bound* é estudada por vários pesquisadores, dentre eles Carlier e Pinson (1989), Applegate e Cook (1991), Brucker et al (1994), entre outros.

Candido (1997), reforçando Wall (1996), cita que esta técnica pode operar em conjunto com a representação por grafos disjuntivos, objetivando solucionar problemas pequenos em tempos razoáveis de tempo, mas falham na solução de problemas maiores, considerando também o tempo consumido.

## 4.2. TÉCNICAS DE SOLUÇÃO APROXIMADAS

Os métodos exatos são técnicas garantidas que permitem encontrar, caso existam, soluções ótimas. Já os métodos aproximados ou heurísticos, apesar de não darem tais garantias, possibilitam encontrar outras soluções bastante próximas das primeiras. Esta desvantagem é compensada pelos tempos computacionais, adequados a grandes problemas e que poderiam tornar-se impraticáveis nas técnicas exatas.

O termo heurística origina-se do grego *heuriskein* e que significa achar, encontrar. Em pesquisa operacional, como define Goldbarg e Luna (2000), uma heurística é uma técnica que permite obter boas soluções com esforço computacional razoável, capaz de garantir a viabilidade ou a otimalidade da solução encontrada, ou ambas. De acordo com os autores, as primeiras técnicas heurísticas surgiram associadas a problemas enumerativos, soluções parciais ou relaxações.

Dada a natureza NP - completa dos JSSP, encontra-se na literatura diversos trabalhos envolvendo métodos heurísticos. Glover e Greenberg (1989) apud Jain e Meeran (1998) reforçam esta idéia quando consideram que métodos exatos de buscas em árvores são insatisfatórios para problemas com dificuldades combinatórias, e salientam que métodos inspirados em fenômenos naturais ou por inteligência são mais

susceptíveis a promover a ligação bilateral entre pesquisa operacional e inteligência artificial.

Nesta tese, as heurísticas associadas aos agendamentos de tarefas vêm por atuar na escolha de uma operação entre outras concorrentes, mediante certo(s) critério(s). Caso a tarefa possa ser executada em mais de um recurso produtivo, a heurística poderá contribuir com esta escolha, observadas as restrições de designação de tarefas a recursos, intrínsecas ao problema. Esta característica é a *flexibilidade de rotas* doravante mencionada.

De forma mais elucidativa e apoiando-se na literatura que envolve o tema, a seguir serão apresentadas algumas técnicas aproximadas aplicadas na solução de problemas de agendamento de operações em *job-shops*. Partindo de métodos simples e tradicionais como as regras de despacho, também outras técnicas como *simulated annealing* e *busca tabu* serão elencados. Os *algoritmos genéticos*, peça chave deste trabalho, também serão evidenciados.

#### 4.2.1. Regras de despacho

As regras de prioridade de despacho, ou *Priority Dispatching Rules (PDR's)*, são provavelmente as técnicas heurísticas mais populares aplicadas aos problemas de agendamento de tarefas. Aplicadas nos trabalhos de Jackson (1955) e Giffler e Thompson (1960), têm sua importância evidenciada em diversos trabalhos ao longo do tempo e, mesmo na atualidade, ainda são muito utilizadas em métodos combinados. Exemplos disto são os trabalhos de Singh, Mehta e Jain (2006), El-bouri e Shah (2006), Natarajan et al (2007), dentre outros.

Sua importância reside na fácil implementação e no reduzido custo computacional necessário. Em linhas gerais, o procedimento consiste em escolher operações de um conjunto, ainda não escalonadas. Segundo um critério de escolha (tempo de finalização mais cedo, por exemplo), localiza-se as operações que utilizam uma dada máquina e, mediante o tempo de disponibilidade desta e das operações que podem ser nela executadas a começar o mais cedo possível, são postas em um conjunto de conflito. Uma operação será selecionada deste conjunto pelo critério adotado, e posteriormente inserida no sequenciamento. Quando todas as operações forem seqüenciadas, o procedimento para. O referido critério de seleção é que irá definir a regra de prioridade aplicável ao caso. No texto, mencionou-se a de menor tempo de execução, que será denotada pela regra *Shortest Processing Time - SPT*, a ser evidenciada posteriormente. Esta regra foi também aplicada por Branco et al (2007)

no estudo de convergência de soluções em algoritmos genéticos aplicados aos JSSPs clássicos com populações iniciais criadas por métodos gulosos (SPT).

As PDR's podem ser classificadas como regras globais ou locais. São locais quando definem a próxima operação a ser inserida em uma sequência pela observação de cada máquina, individualmente. Quando se baseiam em toda a produção a ser feita para compor a indicação da próxima operação, são ditas globais.

Outra forma de classificá-las é através de categorias, envolvendo: tempos de processo; prazos de entrega; nenhum dos anteriores, mas bastante simples; condições de chão de fábrica; duas ou mais das condições anteriores. Observa-se que não há uma regra única que garanta boa performance em todo e qualquer critério que se possa adotar. Em geral, nota-se que regras baseadas em tempos de processo ajustam-se melhor a condições de carregamento elevado, enquanto que regras baseadas em prazos de entrega atuam melhor em sistemas menos carregados. Então, a escolha de uma PDR deve considerar os critérios que se almeja aprimorar durante no agendamento das operações (RAJENDRAN & HOLTHAUS, 1999).

Alguns estudos presentes na literatura referentes aos JSSP apontam para regras eficientes, para diferentes medições de desempenho. Algumas regras locais, considerando critérios de tempo de processamento, se destacam (RAJENDRAN E HOLTHAUS, 1999; BERTEL E BILLAUT, 2004; MATTFELD E BIERWIRTH 2004; CHIANG E FU, 2006):

- *FIFO (first in, first out – primeiro a entrar, primeiro a sair)*: Regra utilizada como referência na solução de conflitos, dado que despreza propriedades das tarefas e estado das máquinas. Define que a primeira operação oferecida para ingresso na máquina será aceita, em detrimento de outras. Esta regra é eficiente na minimização do máximo tempo de fluxo;

- *SPT (shortest process time - menor tempo de processamento)*: Regra mais comumente usada nos agendamentos *Job-shop*, visa dar maior prioridade a tarefa que apresentar menor tempo de operação, para a mesma máquina vigente. É classificada como bastante eficiente na minimização do tempo médio de fluxo (especialmente em chão de fábrica sob condições elevadas de carregamento), além de sua também eficiência na minimização de proporções de operações atrasadas;

Deve-se salientar que existem diversas outras regras envolvendo

tempos de processamento, como as SRPT, LTWK e SPT/TWK, comentadas por Chiang e Fu (2006) e relacionadas na Tabela 4.1. Tais regras, extensões da SPT, aliam, sob condições combinadas, outros objetivos, também com boa eficiência quanto às operações atrasadas.

Ainda tocante aos objetivos baseados em tempos de processamento, Rajendran e Holthaus (1999) listam a regra LPT (*longest processing time* – tempo de processamento maior), que se assemelha à SPT, a não ser pelo fato de conceder maior prioridade a operações com maior tempo de processamento. Jain e Meeran (1998) apud Chang et al (1996) apresentam que, em um estudo avaliando 42 PDR's em modelo de programação linear inteira, a regra SPT demonstrou melhor desempenho e a LPT, contudo, obteve a pior média.

No tocante a datas devidas, diversos outros trabalhos são apresentados, evidenciando regras tomadas como referências. A seguir são apresentadas algumas notações utilizadas nas equações matemáticas das regras a serem discutidas posteriormente:

$Z_j$ : índice de prioridade da tarefa  $J$ ;  
 $t$ : instante do tempo em que uma operação é alocada a um recurso produtivo;  
 $i$ : índice da operação;  
 $j$ : índice da tarefa (*job*);  
 $p_{ij}$ : tempo de processamento da operação  $i$  da tarefa  $j$ ;  
 $k$ :  $k$ -ésima operação da tarefa  $j$ ;  
 $r$ : tempo requerido pelas oper. restantes de  $j$ , a partir de  $t$ , considerando  $p$ ;  
 $s_j$ : folga da tarefa  $j$  ( $s = d - t - r$ );  
 $s'$ : folga da operação, sendo  $s' = d - c.(r - p)$ , sendo  $c$  = parâmetro;  
 $O$ : número total de operações de cada tarefa;  
 $e_j$ : instante de chegada da tarefa  $j$  ao sistema;  
 $d_j$ : instante ou data de entrega da tarefa  $j$ ;  
 $d_{jk}$ : data de entrega da operação  $k$  na tarefa  $j$ ;  
 $tp$ : média do tempo de processamento da operação, na máquina;  
 $tp'$ : tempo total de processamento da operação, na próxima máquina;  
 $w_j$ : soma dos tempos de espera das operações ainda não completadas na tarefa  $j$ ;

Listadas a seguir, tem-se algumas regras que envolvem especificamente os objetivos de prazos de entrega (RAJENDRAN E HOLTHAUS, 1999; BERTEL E BILLAUT, 2004; MATTFELD E

BIERWIRTH 2004; ARAUJO JR, 2006; CHIANG E FU, 2006)

- *EDD (earliest due-date – prazo de entrega mais cedo)*: Esta regra é comumente utilizada pela indústria pela sua simplicidade de aplicação no chão-de-fábrica, selecionando a operação que pertencente à tarefa com data de entrega mais cedo. Apresenta boa *performance* na minimização de atrasos máximos e do número de operações atrasadas;

- *ODD (operation due-date – data de entrega de operação)*: Regra que prioriza a operação com data de entrega mais cedo. Apresenta boa *performance* na minimização de atrasos máximos. A data de entrega de uma operação é encontrada da seguinte forma:

$$\text{Seja o subintervalo: } (d_j - e_j) / O' \quad (4.1)$$

Então faça  $d_{kj} = k$  subintervalo, para  $k=1, \dots, O$ .

A expressão mostra que a data de entrega de certa operação da tarefa é obtida dividindo-se o intervalo existente entre data devida  $D_j$  da tarefa subtraído a data de entrada  $E_j$ , e posteriormente dividindo-se o valor encontrado pela quantidade de operações  $M'$  que a tarefa apresenta. O final de cada operação é estimado por cada data final de cada subintervalo;

- *MDD (modified due-date – data de entrega modificada)*: Esta regra prioriza a operação associada à tarefa que possui a data de entrega modificada mais cedo. Define-se a data de entrega modificada da tarefa  $j$  da seguinte forma:

$$d_j' = \max \{d_j, t + r\} \quad (4.2)$$

Observa-se que se a tarefa  $j$  atrasar, a nova data de entrega ( $d_j'$ ) será seu instante mais cedo possível de término. Caso contrário, a data permanecerá a original. É uma regra que combina o princípio da EDD com a SRPT, que visa priorizar operações com menor tempo remanescente, a partir de  $t$  (vide Tabela 4.1). Sua *performance* é eficiente para casos envolvendo atrasos médios.

- *COVERT (cost over time – custo sobre tempo)*: Eficiente na medição de *performance* em avaliações relacionadas a atrasos nos *job-shops*, esta regra calcula uma função de penalidade  $C_i$ , dependendo da folga da tarefa  $j$  para então calcular o índice de prioridade  $Z_j$ . A tarefa com maior  $Z_j$  é a escolhida:

$$Z_j = c_j / p_{ij}, \quad (4.3)$$

onde

$$c_i = \begin{cases} (WT - s) / WT, & \text{se } 0 \leq s_j < WT_j \\ 0 & , \text{ se } s_j \geq WT_i \\ 1 & , \text{ se } s_j < 0 \end{cases} \quad (4.4)$$

- *S/RPT (slack per remaining processing time – folga por tempo de processamento restante)*: Regra que prioriza operações com base no índice composto pela razão entre a data de entrega menos o tempo de término da tarefa e o tempo de processamento restante da tarefa.

- *WINQ (work in next queue – trabalho na próxima fila)*: Esta regra prioriza, dentre as operações das tarefas concorrentes a um recurso m, qual o menor índice de trabalho total para as operações subseqüentes destas tarefas nos recursos imediatamente seguintes ao processamento feito por m. Em suma, prioriza os menores valores de  $l'$ .

Alguns pesquisadores sugerem que a idéia de combinar regras de prioridade simples para formarem outras mais complexas podem ser interessantes. Dois são os procedimentos padrões: métodos aditivos e métodos alternativos. As aditivas ou por peso, determinam a prioridade pelo cálculo da expressão (HOLTHAUS & RAJENDRAN, 1997):

$$Z_j = \sum_{f=1}^g \alpha_f \cdot (Q_f)_j \quad (4.5)$$

Onde:

$(Q_f)_j$  é o valor de prioridade da regra  $f$  para a tarefa  $j \mid f = 1, 2, \dots, g$ ;

$\alpha_f$  é o coeficiente, ou peso, da regra  $f$ , onde  $\alpha_f > 0$ ;

$Z_j$  é o índice resultante da prioridade para a tarefa  $j$ .

Uma desvantagem neste método é que  $Z_i$  fica sensível aos valores de  $\alpha_f$  e conforme O'grady e Harrison (1985) necessitam de algoritmos para encontrar seus melhores valores. Os dados utilizados no



cálculo de  $\alpha_f$  implicam num comportamento do chão de fábrica igual pra o futuro, o que pode não ser verdade. Métodos hierárquicos, ao contrário, baseiam-se em procedimentos condicionais e empregam apenas duas regras simples combinadas numa única expressão.

Holthaus e Rajendran (1997) desenvolveram regras baseadas em ambas as técnicas, envolvendo regras como a SPT e a WINQ. A motivação em utilizá-las combinadas é que a primeira é eficiente nos tempos médios de processamento e a WINQ pode auxiliar na redução dos tempos de espera das tarefas, uma vez que se utiliza de informações do chão de fábrica, tendendo a dar preferências a tarefas que se movem em filas e que podem parar em futuros congestionamentos.

A Tabela 4.1, a seguir, vem por apresentar, de forma resumida, algumas destas PDR's e outras, igualmente eficientes ou comumente utilizadas, dispostas em estudos recentes. Basicamente apresentam objetivos de tempos de fluxo ou datas devidas e, para cada regra é possível observar as equações matemáticas de cálculo dos índices de prioridade. O símbolo de maior ou menor indica a forma como a priorização é realizada, ou seja, se escolhida à operação de maior ou menor índice, respectivamente. As setas denotam a forma de escolha, pela regra, dos valores correspondentes aos fatores das colunas.

PDR	Z =	Prioriz.	P	r	P	q	d	s	s'	l'
SPT	P	<	↓							
SRPT	R	<		↓						
LTWK	P	<			↓					
SPT/TWK	p/P	<	↓		↑					
SLACK	S	<						↓		
EDD	D	<					↓			
MDD	Max{d, t+r}	<		↓			↓			
ODD	s'	<							↓	
MOD	Max{s', t+r}	<		↓					↓	
COVERT	$(1/p) \cdot (1-s/k_b \cdot (r-p))$ *	>	↓	↑				↓		
ATC	$(1/p) \cdot \exp(-(s/k_a \cdot l))$ *	>	↓					↓		
CR	r/(d-t)	>		↑			↓			
PT+PW	p+q	<	↓			↓				
PT+PW+ODD	p+q+s'	<	↓			↓			↓	
WINQ	tp'	<								↓
PT+WINK+SLACK	p+tp'+s'	<	↓					↓		↓

\* - os símbolos kb (COVERT) e ka (ATC) são parâmetros das regras.

FONTE: ADAPTADO DE CHIANG E FU (2006).

TABELA 4.1 – RELAÇÃO DE ALGUMAS PDR'S CONSIDERANDO DATAS DEVIDAS E TEMPO DE FLUXO

No tocante aos interesses da tese, torna-se importante para estudo, a consideração de prazos devidos na execução dos processos, dadas às características de demanda orientadas a encomenda/pedidos que os processos estão sujeitos, mas sem esquecer-se das condições pertinentes a tempos de fluxo nos processos, o que conduz a regras combinadas.

Com base nas informações colhidas na literatura, utilizar-se de regras mais simples para efeitos de combinação poderão surtir efeitos melhores no tocante a tempo computacional. Assim sendo, buscar um confronto entre regras combinem características das SPT e S/RPT, apresentada por Araujo (2006) além de outra, que opere com SPT e WINQ serão consideradas. A regra SPT é forte candidata, pois tem boa referência em diversos trabalhos, sendo eficiente na redução de tempo de fluxo de produção. Como também já visto, regras como S/RPT e WINK mostram-se eficientes nos campos de redução de tempos de atraso e processos em fila. Os resultados obtidos desta análise serão muito úteis durante os procedimentos de agendamento, inseridos no algoritmo geral do trabalho. Entende-se que conciliar poucas regras com resultados eficientes individualmente nos campos de menor tempo de fluxo de processo e menores tempos de atraso quanto a prazos de entrega trarão, à luz dos estudos mencionados nesta seção, bons resultados quando combinadas, formando contribuições importantes ao objetivo global do trabalho.

#### 4.2.2. Sistemas de base de conhecimento

Estes sistemas são aptos a operar com instâncias reais de problemas de agendamento, mas apresentam custo elevado de tempo e de desenvolvimento, além de suas performances estarem dependentes das performances dos profissionais especializados (de baixa qualidade, em situações combinatórias). As aplicações são usualmente dependentes dos ambientes e conclusões gerais são difíceis de apresentar (CANDIDO, 1997).

#### 4.2.3. Relaxação lagrangeana

A relaxação lagrangeana (RL) é uma técnica empregada para encontrar limites inferiores, em problemas de otimização combinatória. Basicamente, consiste em relaxar algumas restrições do problema originalmente de programação inteira, inserindo-as como parte da função objetivo, associados aos operadores lagrangeanos.

#### 4.2.4. Metaheurísticas

As metaheurísticas são procedimentos de busca por soluções, consistindo na aplicação, em cada passo, de uma heurística subordinada, a qual tem que ser modelada para cada problema específico. Contrariamente às heurísticas convencionais, as metaheurísticas são de caráter geral e têm condições de escapar de ótimos locais, diferenciando-se entre si pelo mecanismo usado para escapar destes (SOUZA, 2007).

O autor classifica as metaheurísticas, conforme o princípio usado para explorar o espaço de soluções, como sendo: busca local ou busca populacional. No primeiro caso, a exploração é feita por meio de movimentos aplicados a cada passo sobre a solução corrente, gerando outra solução promissora em sua vizinhança. *Busca Tabu* (GLOVER E LAGUNA, 1993) e *Simulated Annealing* (KIRKPATRICK, GELATT E VECCHI, 1983) são exemplos desta classe. Já os métodos baseados em busca populacional, consistem em manter um conjunto de boas soluções e combiná-las de forma a tentar produzir soluções ainda melhores. Exemplos clássicos são os Algoritmos Genéticos (GOLDBERG, 1989) e os Algoritmos Meméticos (MOSCATO, 1999). A seguir serão dispostas informações mais detalhadas de funcionamento de algumas destas metaheurísticas, focando o problema tratado na tese.

##### 4.2.4.1. Busca tabu

Esta metaheurística é considerada uma técnica de otimização global iterativa. Tendo origem a partir da busca por soluções de problemas de programação inteira, posteriormente foi estendida a quase todos os problemas combinatórios (GOLDBARG E LUNA, 2000).

Em linhas gerais, a *Tabu Search* (Busca Tabu – BT) é um procedimento que restringe a busca e procura por otimalidades locais, armazenando a história da busca em memória. Ela proíbe (faz tabu) movimentos na vizinhança com certos atributos, com o objetivo de guiar o processo de busca para além de soluções que (baseadas em informações disponíveis) tenham duplicidade ou assemelhem-se a soluções previamente armazenadas/obtidas. A função de curta duração da memória permite “esquecimentos estratégicos”, para apenas executar os  $t$  mais recentes movimentos tabu. Entretanto o estado tabu de um movimento não é absoluto. O critério de aspiração leva um movimento tabu a ser selecionado se apresentar certo nível de qualidade. Memórias de média e longa duração também podem ser aplicadas a fim de

promover explorações mais amplas do espaço de busca. Estratégias imediatas ou de média tendência são baseadas na modificação das regras de escolha para encorajar movimentos e soluções historicamente tidas boas, retornando para espaços atrativos do domínio da busca e intensificando a procura nestas regiões. Métodos longos diversificam a busca em áreas ainda não exploradas, podendo se basear na modificação de regras para incorporar atributos na solução que não é frequentemente utilizada (JAIN E MEERAN, 1998).

Para a utilização da Busca Tabu, é preciso também definir uma função objetivo do problema. Definida  $f$ , é preciso gerar uma solução inicial viável (normalmente através de regras de despacho). A partir desta iniciar-se-ão as buscas.

A seguir, apresenta-se um algoritmo voltado para escalonamento de tarefas, baseado em *busca tabu* e utilizando a função objetivo  $f$  mencionada.

```

while (niter - melhiter < nbmax) do
  for "m" máquina do
    f' ← fmelhor;
    niter ← melhiter + 1;
    p ← p*;
    Gerar  $V_e^*$  de soluções  $(e, p)_i$  em  $N_e(e, p^*)$  ou  $f((e, p)_i) < A(f(e, p^*))$ ;
    Atualizar Lista Tabu  $L$  e  $A(z)$ ;
    if  $f(e', p^*) < fmelhor$  then
      | fmelhor ←  $f(e', p^*)$ ;
    endif
    Gerar  $V_p^*$  de soluções  $(e, p)_i$  em  $N_p(e, p)$  ou  $f((e, p)_i) < A(f(e, p))$ ;
    Atualizar Lista Tabu  $L$  e  $A(z)$ ;
    if  $f(e', p') < fmelhor$  then
      | fmelhor ←  $f(e', p')$ ;
    endif
    p ← p';
    e ← e';
  endfor
  if fmelhor < f' then
    | melhiter ← niter;
  endif
endw

```

FONTE: MÜLLER E GÓMEZ (2006)

FIGURA 4.1 – ALGORITMO DE BUSCA TABU, MODIFICADO PARA ESCALONAMENTO DE ATIVIDADES

Em linhas gerais, o algoritmo busca, em cada iteração um ótimo local e, a partir dela, gera uma nova vizinhança. Para evitar mínimos locais e ciclos, é implementada uma lista de movimentos proibidos, ou lista tabu. Se a iteração obtiver uma melhora na função  $f$ , utiliza-se a função de critério de aspiração  $A$ , que admite o movimento até então proibido. Para condições de parada do algoritmo, pode-se utilizar um número máximo de iterações ( $nbmax$ ) sem que ocorra melhoria na função  $f$  ou executar até que alcance o valor mínimo, se conhecido

(MÜLLER E GÓMEZ, 2006).

Jain e Meeran (1998) apresentam o método BT como sendo um dos mais eficientes na busca por boas soluções em sistemas *Job-shop* clássicos. Evidenciam diversas citações de melhorias nas estruturas de memória, para gerenciamento das proibições. Salientam ainda que métodos, como *Branch-and-bound*, se combinados a ela, apresentam melhoras nas buscas, contudo com maior custo computacional. Como na maioria das estratégias de busca local, a BT requer muitos parâmetros para operar e estes devem ser cuidadosamente ajustados, e considerando as diversidades de aplicação eminente nos casos reais a serem estudados, isto poderá ser uma barreira difícil de transpassar.

#### 4.2.4.2. Têmpera simulada (*Simulated annealing*)

Pertencente ao ramo das técnicas de busca denominadas *Aleatório-guiadas*, a Têmpera Simulada, ou *Simulated Annealing* – SA, apresentam componentes aleatórios, mas também empregam informações do estado corrente para guiar a pesquisa da solução do problema estudado. É um método de busca local que aceita movimentos de piora para escapar de ótimos locais.

Proposto originalmente por Kirkpatrick et al (1983) se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. Para a utilização do SA, deve-se definir a priori, um método para geração de uma solução inicial  $S$ , um método para geração das soluções vizinhas  $S'$  (estrutura de vizinhança), e uma função objetivo  $f(S)$  a ser otimizada (MAURI E LORENA, 2006).

Algumas contribuições às funções de vizinhança para o JSSP foram postas por Van Laarhoven et al (1992) apud Jain e Meeran (1998). Basicamente consistem em movimentos obtidos pela reversão das ordens de processamento de um par de operações críticas adjacentes, se sujeitas a processamento na mesma máquina. Tal método de SA proposto mostra-se bastante robusto.

Yamada e Nakano (1996) empregam seu método CBSA – *Critical Block Simulated Annealing* (tempera simulada em bloco crítico) – que em linhas gerais agrega uma estrutura de vizinhança derivada de blocos críticos ao SA, e soma a este, o gerador de agendamentos proposto por Giffler e Thompson (1960).

A seguir, descreve-se um algoritmo de SA proposto por Yamamoto (2004), contudo também França (2007) atenta para estes detalhes de desenvolvimento e implantação para funções contínuas.

Yamada e Nakano (1996) propõe exemplos de aplicações em problemas combinatórios de JSSP.

Passo 1: Estima-se uma temperatura inicial  $T$ , permitindo um espaço de busca, de novas soluções, sem limitações.

Através da configuração original obtém-se a energia( $i$ ) inicial.

Passo 2: Enquanto não for alcançado o critério de parada, vá para o Passo3. Alcançado o critério de parada, vá para o Passo8.

Passo 3: Para  $k$  de 1 a  $L$  (número de iterações por nível de temperatura). Para que, de forma gradativa, haja o esfriamento do material até seu estado final de ‘congelamento’, ou seja, haja uma aumento na restrição da variação da pesquisa de uma nova solução, o valor da variável  $L$  é calculada por:  $L = \text{número máximo de iterações} / \text{num. de trechos abertos}$ . Após  $L$  iterações, vá para o Passo7.

Passo 4: Gerar nova configuração de forma aleatória ( $j$ ), sendo uma configuração factível, radial, não permitindo fechar o circuito ou interligar duas subestações. Calcular energia( $j$ ).

Passo 5: Se (energia( $j$ )<energia( $i$ )). Então  $i=j$ , vá para o Passo3.

Caso contrário calcular .

Passo 6: Se ( $p > \text{rand}(0,1)$ ) , onde  $\text{rand}(0,1)$  é um número aleatório uniformemente distribuído entre 0 e 1. Então  $i=j$ , vá para o Passo3. Caso contrário rejeitar  $j$ , vá para o Passo3.

Passo 7: Diminua o valor de  $T$ , de forma a diminuir o espaço de busca das novas soluções. Vá para o Passo2.

Passo8: Pare. A solução ( $i$ ) é a melhor configuração da rede.

Considerando diversos trabalhos envolvendo o SA, além das contribuições ao método associadas, nas aplicações voltadas ao JSSP, Jain e Meeran (1998) salientam que os resultados são pobres. Somente com a incorporação de outras técnicas (ex.: algoritmos genéticos, dentre outros) é que a qualidade dos resultados é melhorada. Os autores mencionam também que a maior deficiência desta técnica reside no excessivo consumo de tempo computacional para que boas soluções possam ser encontradas. Outro grave problema são os parâmetros do SA, altamente dependentes da natureza do algoritmo. Ressaltam ainda que este consumo de tempo computacional esteja associado à necessidade de diversas execuções sobre o problema, para que boas soluções possam ser encontradas. Acrescenta-se que resfriamentos lentos também potencializam a obtenção de melhores resultados, entretanto também geram um consumo de tempo computacional considerável.

#### 4.2.4.3. Algoritmos evolucionários

Diante da sua grande aplicação na solução de problemas de classe NP - completa, os Algoritmos Evolucionários (AEs) compõem técnicas heurísticas de busca baseadas em mecanismos naturais de seleção.

Estes sistemas simulam computacionalmente ambientes baseados nestes princípios de evolução e hereditariedade onde, operando com uma população de estruturas (cromossomos, ou indivíduos) empregam técnicas de seleção orientadas pelas aptidões de cada indivíduo e, posteriormente, operadores genéticos como reprodução e mutação.

Esta “caminhada coletiva de uma população” na busca por melhores soluções é fruto de um aprendizado coletivo onde cada indivíduo visa melhorar sua adequação em relação ao meio ambiente, ou ao problema, se considerada a metáfora (GOLDBERG, 1989).

De acordo com Linden (2008), existem diversos modelos computacionais propostos, baseados no conceito de simulação da evolução das espécies por meio da seleção e operadores de reprodução e mutação, sendo estes, dependentes da aptidão de cada indivíduo, em sua espécie e no ambiente em que está inserido. Tanomaru (1995) e Barboza (2005) citam as Estratégias Evolucionárias (EE), Programação Genética (PG), Sistemas Classificadores (SC's) e a Transgenética Computacional (TC), dentre outros.

Destes métodos, o mais difundido e pesquisado é o Algoritmo Genético (AG), dada sua flexibilidade e eficácia em realizar busca global em ambientes diversos (BARBOZA, 2005).

### 4.3 ALGORITMOS GENÉTICOS APLICADOS AO JSSP

Os primeiros trabalhos nesta linha são originários de Holland (1975), objetivando fundamentar uma teoria geral de sistemas e adaptação robusta, com aplicação prática na determinação de máximos e mínimos de funções matemáticas.

Pertencentes a classe dos AE's, os algoritmos genéticos apresentam características distintas das outras metaheurísticas, a saber: opera em um conjunto de pontos (população) e não a partir de pontos isolados; opera em um espaço de soluções codificadas e não diretamente no espaço de busca; necessita como informação, somente o valor de uma função objetivo (função de adaptabilidade, ou *fitness*); usa transições probabilísticas e não regras determinísticas (GOLDBERG e LUNA, 2000). Ressalta-se que os AG's, apesar da forte componente aleatória, tem suas caminhadas também guiados pelas informações históricas

contidas nas gerações para encontrar novos pontos de busca.

Os procedimentos executados por um AG clássico são, criada a população inicial (cada indivíduo é codificado em uma estrutura denominada cromossomo), calcula-se a adaptação (*fitness*) para cada indivíduo. Operadores genéticos são aplicados a indivíduos selecionados, tradicionalmente com base nos seus *fitness*, e uma nova geração de indivíduos é criada. Contudo, novas gerações também podem ter a coexistência de cromossomos na população corrente. Na Figura 3.6 tem-se o fluxograma elementar de um AG.

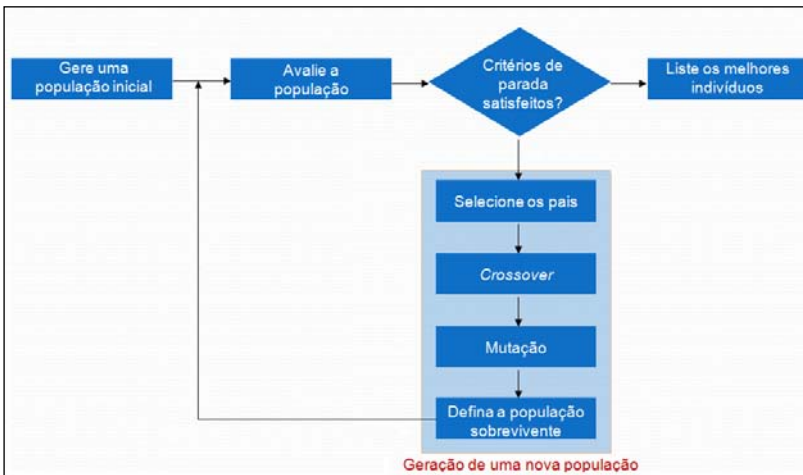


FIGURA 4.2 – FLUXOGRAMA SIMPLIFICADO DE UM ALGORITMO GENÉTICO CLÁSSICO

Os procedimentos são repetidos até que o critério de finalização seja atingido.

Goldberg (1989) evidencia alguns fatores, intrínsecos da estrutura disposta na Figura 3.6, que são fundamentais para definir o desempenho dos AGs nos casos clássicos dos JSSP: geração da população inicial, métodos de seleção, escalas de *fitness*, tamanho de população, critérios de parada, formas de decodificação, nível de hibridização, e outros.

A seguir se discute cada uma destas etapas, elucidando os pontos que motivarão os estudos deste trabalho, envolvendo os AG's, regras heurísticas e sequenciamento de tarefas em ambientes de produção do tipo *job-shop*.

#### 4.3.1. Terminologia



Os algoritmos genéticos empregam uma terminologia advinda das ciências biológicas, mais precisamente da genética, a fonte inspiradora deste método de busca.

O AG atua sobre um conjunto de pontos dispersos em um espaço de busca, denominados indivíduos (ou cromossomos). Seu conjunto é denominado população.

Os cromossomos são formados por *genes* e estes podem assumir valores, denominados *alelos*. Sabe-se ainda que os *alelos* têm posições fixas no cromossomo, também chamadas de *lôcus* (ou *loci*, no plural). A seguir é disposta uma tabela para relacionar os termos mais empregados na genética e também no âmbito dos AGs.

<i>Linguagem natural</i>	<i>AG</i>
Cromossomo	Indivíduo, string, cromossomo
Gene	Característica
Alelo	Valor
Lôcus	Posição
Genótipo	Estrutura
Fenótipo	Conjunto de parâmetros

FONTE: ADAPTADO DE LINDEN (2008) E BARBOZA (2005)

TABELA 4.2 – RELAÇÃO DE TERMINOLOGIAS ASSOCIADAS AOS AG'S

Os termos genótipo e fenótipo são igualmente discutidos ao longo do trabalho e significam a estrutura do cromossomo e sua integração com o problema, respectivamente. Representados sob a forma de cromossomos, os indivíduos normalmente são vetores, onde cada posição (*locus*) denota sua característica: genótipo. Combinados ou alterados, formam a característica geral do indivíduo: fenótipo. Genótipos diferentes podem gerar fenótipos satisfatórios para a sobrevivência do indivíduo.

A seguir é abordada a codificação das soluções em cromossomos: definição dos genótipos e a importância desta etapa na concepção de um AG para um dado problema.

#### 4.3.2. A codificação das soluções: os cromossomos

A escolha da melhor estrutura para representar uma solução nos AGs tem sua problemática associada na forma mais viável de traduzir e tratar computacionalmente as informações do problema. Uma boa representação deve ser simples, não permitir representar soluções proibidas e quanto mais adequada ao problema, maior a qualidade dos resultados obtidos (LINDEN, 2008).

Nos JSSP, se os cromossomos forem representações inapropriadas, o AG poderá apresentar dificuldades para encontrar boas soluções. As representações de natureza combinatória tendem a ser muito difíceis, principalmente porque as operações de reprodução dos AGs podem conduzir a soluções ineficazes (HERRMANN et al, 1995).

Uma representação binária clássica proposta por Yamada (1995), assim como a aplicada por Holland (1975), realiza a construção de soluções através da formação aleatória de *strings* binárias.

Como nos JSSPs cada indivíduo representa um agendamento de operações em recursos produtivos, é preciso observar que este método construtivo pode permitir gerar indivíduos em desacordo com as seqüências tecnológicas definidas para o problema, ou seja, indivíduos “ilegais”. Igualmente, uma formação “factível” irá corresponder a um agendamento executável.

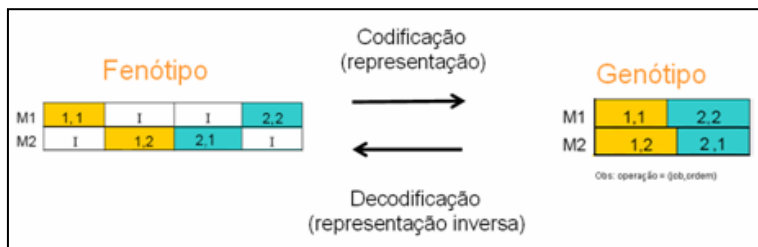
Métodos de harmonização podem ser aplicados como forma de corrigir os indivíduos “ilegais”. Yamada e Nakano (1997) apresentam técnicas de harmonização para tal feito. Em linhas gerais a técnica divide-se em duas fases, sendo a primeira para remover quaisquer inconsistências relacionadas em cada máquina e a segunda, atuando sobre inconsistências globais, ou seja, entre as máquinas.

Além de representações clássicas (binárias), diversas outras são utilizadas, empregando códigos mais complexos. Outra forma, simbólica, considera o cromossomo como sendo uma lista de operações a ser executada. Como cada *locus* define uma tarefa, sua dimensão é definida pelo produto de tarefas e máquinas (  $\dim [j\text{-tarefas} \times m\text{-máquinas}]$  ). A tarefa descrita no primeiro *alelo* é escalonada primeiro; a do segundo, idem, guardadas as condições de precedência entre operações de mesma tarefa. Observa-se que todas as escalas serão válidas e ativas (WALL, 1996).

Na Figura 3.2 (b), como na Figura 3.8, a seguir, observa-se um exemplo de representação simbólica, onde as tarefas são organizadas em um vetor *m*-particionado. De forma semelhante ao apresentado anteriormente por Wall (1996) a estrutura tem as tarefas ordenadas para execução, para cada *m*-máquina. O que difere é que, estando o vetor particionado para máquinas, as tarefas aparecem na ordem a serem executadas no recurso.

Na Figura 4.3, tem-se o genótipo (seqüência de operações) e o fenótipo (agendamento) para um exemplo de solução de um JSSP 2x2. Nela, cada posição do cromossomo tem descrita a [tarefa, operação]. Com o fenótipo definido é possível, por exemplo, saber o *makespan* do

indivíduo (cromossomo).



FONTE: ADAPTADO DE SCOFIELD (2002)

FIGURA 4.3 – EXEMPLO DE CODIFICAÇÃO DE SOLUÇÕES DE JSSP EM ALGORITMOS GENÉTICOS

Cabe ressaltar que as representações estão fortemente ligadas à maneira como os operadores genéticos irão proceder e ainda que nenhuma forma de codificação funciona igualmente bem para todas as situações. Escolher cuidadosamente a forma de como codificar uma solução é crucial para a solução do problema (CASTRO, 1999).

#### 4.3.3. A população inicial

Como já exposto, uma população, para um algoritmo genético, é a concretização do espaço de busca que está sendo pesquisado. Desta forma, a população inicial conterá as soluções a partir das quais se darão as evoluções na busca por melhores resultados.

Dispondo de um espaço de busca, o mais desejável é que os indivíduos estejam dispersos de forma a cobri-lo o mais uniformemente possível. Isso só seria possível para um número infinito de cromossomos. Em termos práticos, uma forma bastante usual de contornar este problema com populações finitas é a de formar indivíduos aleatoriamente, onde se obtém uma boa distribuição de soluções no o espaço de busca.

A população inicial pode, contudo, ser formada empregando técnicas heurísticas. Desta forma, indivíduos de boa qualidade são gerados, auxiliando o AG na velocidade de convergência. Em contrapartida, a população vem a ficar mais homogênea e assim, pode-se ter uma convergência prematura. A exemplo, tem-se o trabalho de Branco et al (2009) que apresenta um estudo voltado aos JSSP's clássicos, analisando a performance de AG's para algumas formações de

JSSP cujas populações iniciais são formadas pelas regras de prioridade de despacho: SPT, RN, e ambas combinadas. O autor salienta que é desejável ter uma parte da população com este tipo de formação, mas que a aleatoriedade é importante para garantir a diversidade da população e não ter convergências prematuras. Quando muito homogêneas é necessário maior uso do operador mutação, que onera de certa forma o processamento geral do algoritmo, haja visto que precisa modificar os cromossomos.

Salienta-se que o operador de mutação colabora efetivamente para uma melhor exploração do espaço de busca, promovendo saltos aleatórios e escapando de regiões com máximos locais.

#### 4.3.4. A função de avaliação

A função de aptidão, ou avaliação, é a forma empregada nos AGs para avaliar o grau de adaptação do indivíduo na população. Em geral, esta qualidade do indivíduo reflete na sua probabilidade de seleção para cruzamento, a ser melhor discutida posteriormente.

Esta função de avaliação retorna um valor numérico associado aos parâmetros do cromossomo e ao problema, denominado *fitness*.

Os AGs são, naturalmente, técnicas de maximização que visam obter a melhor solução, então, para o melhor indivíduo a função aptidão deve retornar o maior *fitness*. Caso o objetivo seja minimizar, a aptidão deverá ser inversamente proporcional ao valor da função.

Nos JSSP, o objetivo clássico é minimizar o menor tempo total de processamento, logo é preciso ajustar a função aptidão, compatibilizando-a com a filosofia dos AGs.

O cálculo dos tempos de processamento está ligado ao agendamento das tarefas, conforme cada indivíduo e alguns parâmetros, como o tipo de agendamento, por exemplo. Definidas as estratégias de cálculo, a forma mais simples pode ser encontrar o inverso do tempo de processamento da sequência denotada pelo cromossomo  $x$ , ou seja:

$$f(x) = 1/tp_x, \quad (4.6)$$

onde  $tp_x$  é obtido pelo agendamento das tarefas conforme rotas e seqüenciamento dados pelo cromossomo.

Observa-se que por ser inversamente proporcional ao tempo de processamento dado pelo agendamento, os indivíduos que denotam maior tempo na execução das tarefas serão menos desejados que aqueles

com menores tempos de processamento. Isso fica expresso na forma do *fitness*, maior para aqueles com menor tempo de processamento.

É importante ressaltar que os *fitness* dos indivíduos não podem ser negativos nem nulos (estritamente positivos), haja visto corresponderem às chances de serem selecionados para reprodução. Uma solução nos casos de números negativos implica na soma de um *offset* a toda série, tornando o menor número também positivo e maior que zero.

#### 4.3.5. O processo de seleção

Como os AGs são inspirados nos mecanismos da seleção natural, deve-se, da mesma forma, proporcionar que os pais mais aptos terão mais chances de perpetuar seus genes por novas gerações mediante o cruzamento. É igualmente importante que os pais menos aptos (com *fitness* muito baixos) também tenham chances de seleção que, embora baixas, não os elimina da evolução. Observa-se que, apesar de menos aptos, sua existência promove a diversidade genética das populações, reduzindo a convergência prematura. Embora seus genes não gerem bons *fitness*, podem ainda colaborar com boas sequencia, quiçá na geração de um superindivíduo, por exemplo.

O método da seleção tem sua importância, pois atua indiretamente na convergência genética das soluções, podendo acelerá-la ou retardá-la, afetando o quão agressivo será o AG no aproveitamento das melhores soluções (LINDEN, 2008).

Nesta fase do AG, é obviamente tentador manter os bons indivíduos para compor gerações futuras. Métodos de seleção com tais características são definidos por Goldberg (1989) como *elitistas*, e embora garantam a convergência, esta pode não ser para um ponto ótimo e sim um máximo local. Uma forma de tentar escapar destas “armadilhas” é através do operador de mutação, ao possibilitar “saltos” aleatórios no espaço de busca.

Tocante a imposição feita pelos métodos de seleção para perpetuar os melhores indivíduos nas gerações futuras, denominada *pressão seletiva*, pode-se medir seu efeito através do quanto uma população têm melhorado ao longo das gerações. O cálculo consiste em comparar a população atual e a anterior, por meio da diferença das médias de aptidão de cada uma, normalizada pelo desvio padrão. Esta *intensidade de seleção* tende a diminuir ao longo das evoluções da população, dada a convergência genética (LINDEN, 2008).

Em domínios epistáticos (JSSP), o uso de relações com maior

pressão na seleção (como a elitista, por exemplo), torna-se mais adequadas (MATTFELD, 1996). Gonçalves et al (2005) reforçam a vantagem da estratégia elitista sobre a tradicional reprodução probabilística, considerando que a melhor solução tende a aprimorar-se de uma geração para a outra. Observa-se que a estratégia *elitista* pode vir inserida em um método de seleção.

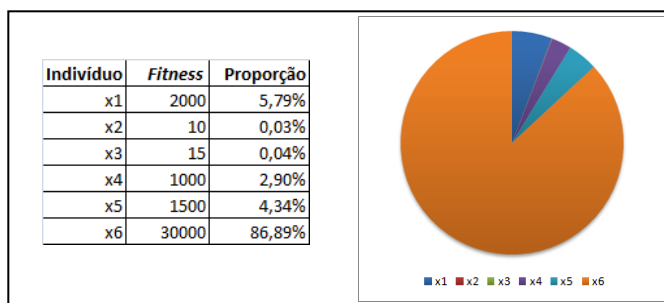
A seguir são apresentadas outras formas conhecidas de direcionar o processo de busca promovendo a seleção dos indivíduos para a reprodução (MITCHELL, 1996).

#### 4.3.5.1. Seleção por roleta viciada

Este processo de seleção, também conhecida por *seleção por Monte Carlo* é feito por *amostragem aleatória*, ou seja, escolhem-se com probabilidade proporcional ao *fitness* de cada indivíduo da população, os “pais” para cruzamento. O cálculo da parcela de cada indivíduo na “roleta” é dado pela Eq. 7, que relaciona a adaptação de cada indivíduo com o *fitness* total da população.

$$\begin{aligned}
 n_{ind} &= \text{número de indivíduos da população} \\
 fitness\_total &= \sum_{i=1}^{n_{ind}} fitness_i \\
 prob\_sel_i &= \frac{fitness_i}{fitness\_total}
 \end{aligned} \tag{4.7}$$

Na Figura 4.4 tem-se um exemplo de *roleta* organizada em função dos *fitness* de cada indivíduo.



FONTE: PRÓPRIA

FIGURA 4.4 – EXEMPLO DE ROLETA VICIADA PARA 20 INDIVÍDUOS

A quantidade de sorteios está associada com a estratégia de

formação da nova população. Se toda a população for submetida a cruzamento, então a quantidade de sorteios será igual à dimensão da população. Há casos onde apenas uma parcela a população é submetida a cruzamentos.

#### 4.3.5.2. Seleção por torneio

Neste método, são escolhidos  $k$  indivíduos de forma completamente aleatória para competirem entre si, objetivando definir quem será o pai, mediante critério do *fitness*.

Tocante ao tamanho do conjunto dos  $k$  indivíduos, observa-se que é preciso um mínimo de dois indivíduos a compor este conjunto, a fim de poder haver competição.

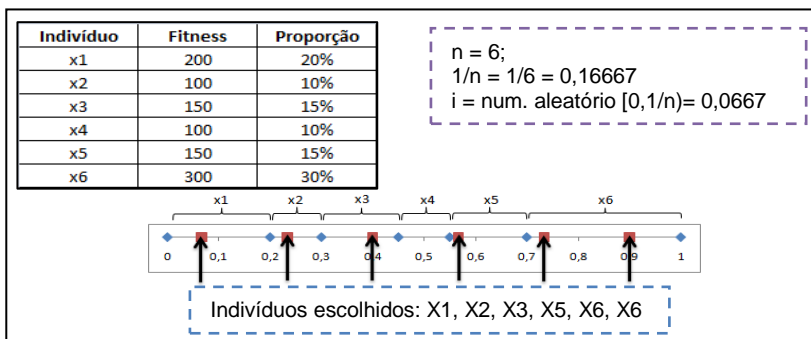
Depois de selecionados, os indivíduos “pais” seguem para o cruzamento. Importante mencionar que neste método, dado que cada indivíduo tem chances iguais de participar do torneio, para o caso de um indivíduo com aptidão ruim, este só terá chances de cruzamento se competir consigo mesmo, logo com chances de  $1/(m^k)$ , onde  $m$  é o tamanho da população. Se  $k$  for alto, estas chances tendem a ser nulas. Isto reflete na questão da diversidade da população. Num torneio de tamanho  $k=5$ , cerca de metade da população é perdida a cada geração. Já empiricamente, com  $k=2$  obtêm-se bons resultados, que inclusive superam os da roleta viciada, sendo pouco sensível a escala da função avaliação, superindivíduos, etc. (LINDEN,2008; BICKLE, 1997).

#### 4.3.5.2. Seleção amostragem estocástica uniforme

Tomando como base o método da roleta, onde os indivíduos têm frações desta conforme suas aptidões, o que difere é no critério de escolha. Ao invés de gerado um número aleatório associado ao comprimento da circunferência, são gerados  $n$  indicadores.

Sabendo-se que  $n$  é também a quantidade de indivíduos a selecionar para cruzamento, define-se que tais indicadores são igualmente espaçados, cobrindo um espaço linear de  $i$  até  $i+(n-1)/n$ . O valor do ponto inicial  $i$  é definido por sorteio, de 0 a  $1/n$ .

Este método não evita a dominância de superindivíduos, além de excluir rapidamente aqueles com baixos *fitness*. Desta forma, a convergência fica muito parecida com a do método da roleta (LINDEN, 2008).



FONTE: ADAPTADO DE LINDEN (2008)

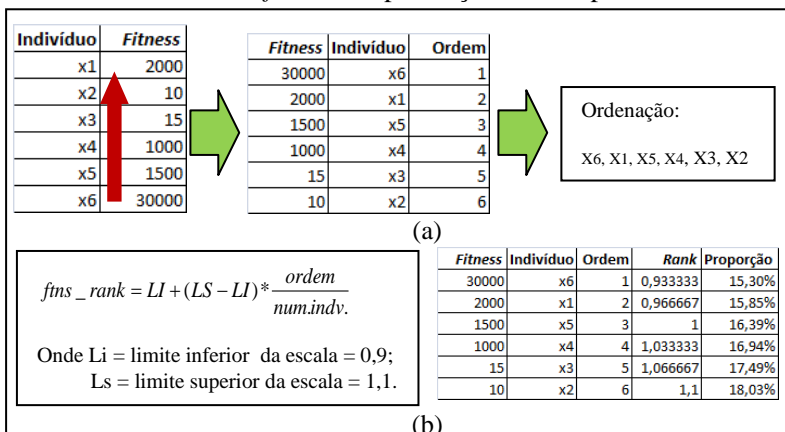
FIGURA 4.5 – EXEMPLO DE SELEÇÃO POR AMOSTRAGEM ESTOCÁSTICA

#### 4.3.5.3. Seleção por ordenação (*ranking*)

Esta técnica organiza os indivíduos ordenando-os conforme seu desempenho. Após, critério de seleção deixa de ser *fitness* e passa a ser a ordem que ocupam, reduzindo a pressão seletiva, mas evitando a convergência prematura e a dominância de superindivíduos.

Salienta-se que o processo de ordenação é bastante oneroso computacionalmente, com complexidade  $O(\log n)$  operações.

A Figura 4.6 compara o método *Monte Carlo* e o por ordenação, com valores diversos de *fitness* e a presença de um superindivíduo.



FONTE: ADAPTADO DE LINDEN (2008)

FIGURA 4.6 – EXEMPLOS: (A) ORDENAÇÃO DE INDIVÍDUOS; (B) PROPORÇÕES PARA ROLETA VICIADA



Linden (2008) promove a diminuição da variância das aptidões dos cromossomos pela ordenação e a linearização dos resultados (entre os valores empíricos 0,9 e 1,1). Sendo seleção por roleta viciada, o indivíduo com *fitness* 30000 teria quase 87% de chances de ser escolhido. Após o *rankeamento*, fica com apenas 15,3%, e os piores e os melhores indivíduos ficam com chances próximas de seleção, aumentando a diversidade da população, reduzindo a convergência genética em uma busca mais lenta no espaço de soluções, que pode até não encontrar máximos locais. Sugere-se, nestes casos, procedimentos combinados com *elitismo*.

#### 4.3.5.4. Seleção de Mayerle

É um método de seleção ordenada estocástica, onde Mayerle (1994) propõe, nos casos de maximização, ordenar os indivíduos de forma decrescente segundo suas aptidões (nos de minimização a ordenação é crescente). O método considera uma distribuição de probabilidade inversamente proporcional ao *rank* dos indivíduos, ou seja, quanto menor o índice, maior as chances de escolha.

O autor define a formula do operador de seleção como segue:

$$posição = Seleção(R) = \left\{ r_j \in P \left| j = p + 1 - \left\lceil \frac{-1 + \sqrt{1 + 4 \cdot rnd(p^2 + p)}}{2} \right\rceil \right. \right\} \quad (4.8)$$

Onde:

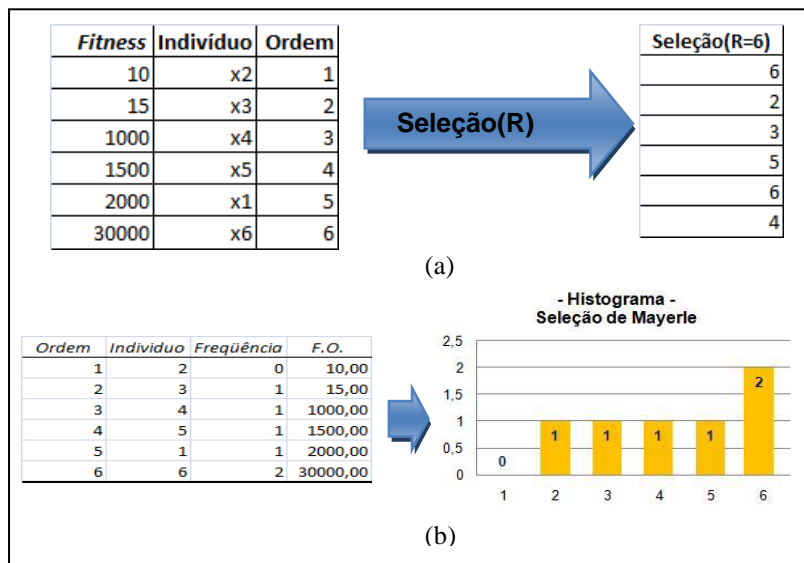
$P$  é o conjunto dos  $p$  indivíduos;

$r_j$  é o  $j$ -ésimo cromossomo;

$rnd$  é um número aleatório uniformemente distribuído  $\in [0,1)$ ;

$\lceil x \rceil$  retorna o menor inteiro maior que  $x$ .

No exemplo, tem-se a seleção para os indivíduos apresentados na Figura 4.4 e 4.6.



FONTES: PRÓPRIA

FIGURA 4.7 – EXEMPLOS: (A) SELEÇÃO DE MAYERLE (1994); (B) HISTOGRAMA DOS SELECIONADOS PARA CRUZAMENTO

O método apresenta uma pressão seletiva maior que o método por ordenação visto na seção anterior, contudo inferior ao da seleção Monte Carlo (ou da roleta viciada). Como resultado, aqueles indivíduos mais aptos tem maior chance de cruzamento que os menos aptos, contudo o surgimento dos superindivíduos é amenizado.

Apesar de necessitar de uma ordenação para posterior seleção, o método não emprega nenhum outro cálculo senão o da Eq. 4.5. Se  $n$  for o número de indivíduos para seleção, também  $n$  será o número de execuções desta equação.

#### 4.3.6. Os operadores genéticos

Os operadores genéticos têm o objetivo principal de promover mudanças na população através das gerações, sendo necessários no processo de diversificação da população, bem como de propagação dos melhores genes ao longo das gerações.

Os operadores mais comuns são o *crossover* e a mutação, apesar de que qualquer método que produza filhos com base em pai(s) ser aceito como operador genético (ex: operador elitista de clonagem, que

copiar um pai de uma geração  $G$ , para uma geração  $G+1$ ).

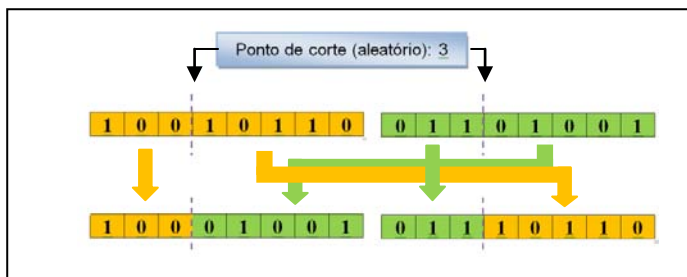
Os operadores genéticos estão fortemente relacionados aos métodos de representação adotados ao sistema do AG. Goldberg (1989) e Yamada e Nakano (1997) evidenciam vários operadores de *crossover* utilizados com o propósito de troca de informações durante formação de novos indivíduos da população. O operador mutação, também utilizado, tendo um papel secundário no processo, geralmente é associado a trocas aleatórias de informação, ou seja, atua na permuta de informações.

Os genes, combinados por estes operadores, tendem a formar seres mais aptos e assim, a evolução natural caminha, com maior adaptação ao meio onde os organismos estão submetidos (LINDEN, 2008).

#### 4.3.6.1. Operador genético de cruzamento (*crossover*)

O *crossover* é o operador genético que tem a função de propagar os genes dos indivíduos mais aptos. Atuando na troca de informações entre os indivíduos pais, o operador de cruzamento simula o fenômeno biológico de *crossing-over*, onde um cromossomo se cruza sobre o outro, trocando segmentos entre si.

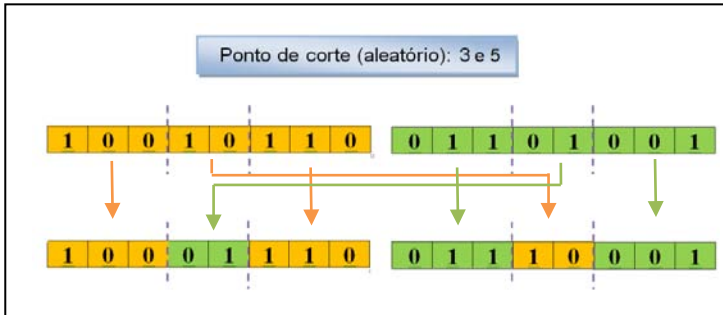
O operador de cruzamento de um ponto é o mais comum e o mais empregado nos AGs, consistindo na divisão do cromossomo em duas partes, não necessariamente iguais, a fim de que se proceda à troca genética de informações. O ponto de corte é definido aleatoriamente e, no processo de geração dos filhos, os pais fornecem de forma complementar seus segmentos, conforme mostra a Figura 4.8.



FONTE: PRÓPRIA

FIGURA 4.8 – EXEMPLO DE OPERADOR DE CRUZAMENTO DE UM PONTO

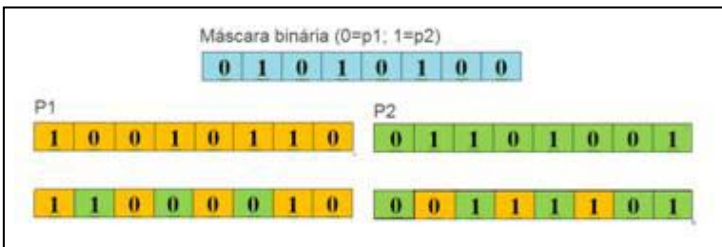
Observa-se que há a possibilidade de cruzamento de múltiplos pontos, onde ocorrem mais de uma troca entre os pais. Este processo de recombinação é mais parecido com o que ocorre de fato na natureza.



FONTE: PRÓPRIA

FIGURA 4.8 – EXEMPLO DE OPERADOR DE CRUZAMENTO DE DOIS PONTOS

Outro operador de *crossover* é o uniforme, onde uma máscara binária guiará a origem dos fragmentos na formação dos filhos. Se zero, a origem é do pai 1. Se um, a origem é do pai 2. Este método é uma generalização dos operadores anteriores.



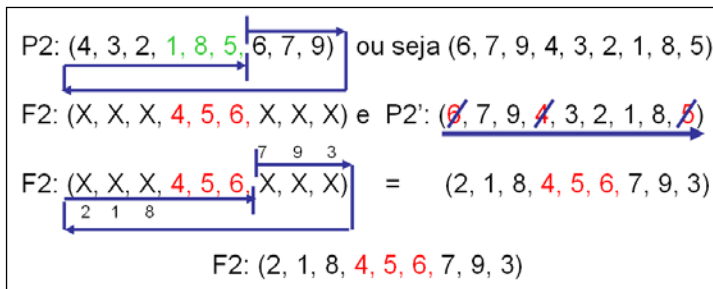
FONTE: PRÓPRIA

FIGURA 4.8 – EXEMPLO DE OPERADOR DE CRUZAMENTO UNIFORME

De acordo com Goldberg (1989), os problemas combinatórios geralmente apresentam elevado grau de *epístase*. Esta ocorre quando mudanças nos genes em diferentes *loci* não são independentes. Problemas *epistáticos* requerem operadores *crossover* não-padrões capazes de produzir filhos viáveis e manter blocos importantes de *loci*

vinculados de forma dependente. Desta forma, elencam-se alguns operadores utilizados nas operações de *crossover* para os JSSP:

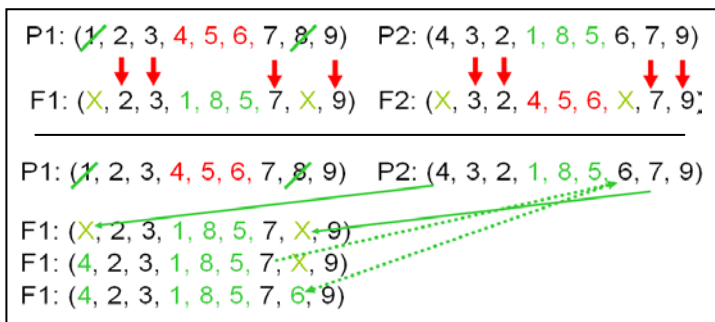
- OX (*ordered crossover* – cruzamento ordenado): utilizado em problemas combinatórios, como o do PCV, este operador gera dois filhos, que herdam a ordem de visita dos pais. Isto ocorre da seguinte forma: a partir do segundo ponto de corte de um dado pai, descreve-se a sequência e posteriormente remove-se, desta, os vértices existentes entre os cortes do filho. Na Figura 2.9 tem-se um exemplo de como isso ocorre (GOLDBARG E LUNA, 2000);



FONTE: ADAPTADO DE GOLDBARG E LUNA (2000)

FIGURA 4.9 – EXEMPLO DE GERAÇÃO DE UM DOS DOIS FILHOS COM OPERADOR DE CROSSOVER OX.

- PMX (*partially matched crossover* - cruzamento parcialmente combinado): como no caso do operador OX, também o PMX necessita dispor de duas *strings* pais. Este procedimento faz com que os filhos herdem a sequência presente entre os dois pontos de corte.



FONTE: ADAPTADO DE GOLDBARG E LUNA (2000)

FIGURA 4.10 – EXEMPLO DE GERAÇÃO DE UM DOS DOIS FILHOS POR CROSSOVER PMX.

Posteriormente, os *gens* do outro pai são copiados ao filho que está sendo gerado. Em caso de já existirem, devido à herança do entre cortes, o outro doador é consultado. No exemplo da Figura 3.10 tem-se a geração de um dos dois filhos. Observa-se que os valores indicados por X são nós que já foram herdados do outro pai, logo não podem ser repetidos. Como solução, verifica-se o que existe nesta mesma posição, do outro pai doador (inicial). Se for possível, o valor também será herdado. Como na posição 8 do doador P2 existe 7 e este já foi inserido do pai P1, não podendo ser re-inserido, verifica-se em P2, qual valor existe na posição referente a ele (7) do F1: o valor é 6. Sendo que este vértice não existe, é integrado ao cromossomo F1 (GOLDBARG E LUNA, 2000);

- PPX (*precedence preservative crossover* – cruzamento de precedência preservativa): desenvolvido para problemas de roteamento de veículos por Blanton e Wainright (1993) e de *scheduling* por Bierwirth et al (1996), consiste em passar as relações de precedência, dadas por dois pais, a um único filho. Em linhas gerais, dado um vetor de dimensão igual ao dos pais, é aleatoriamente preenchido com valores {0,1}, onde 0 indica o pai 0 e 1, o pai 1. Consideram-se os pais e filho como listas, onde as operações são inseridas e apagadas.

<b>P1:</b> 1 2 3 4 5 6 7 8 9	<b>H:</b> 0 1 0 1 1 0 1 0 1	<b>F:</b> 1 4 2 3 8 5 6 7 9
<b>P2:</b> 4 3 2 1 8 5 6 7 9		

FONTE: ADAPTADO DE BIERWIRTH ET AL (1999)

FIGURA 4.11 - EXEMPLO DE GERAÇÃO DE FILHO COM OPERADOR DE Crossover PPX.

Observa-se que apesar de indicado em um pai, se já estiver alocada a operação, a próxima da direita é considerada.

Gonçalves et al (2005) apontam que operadores de reprodução como o PPX podem surtir bons efeitos quando aplicados no lugar de métodos com um ou dois pontos de corte (caso do PMX, por exemplo).

#### 4.3.6.2. Operador genético de mutação

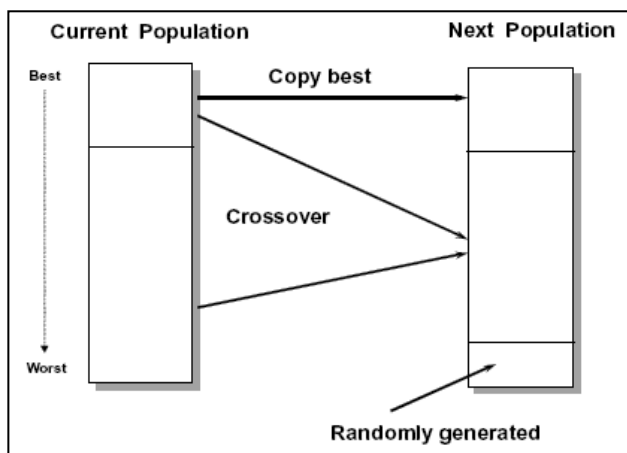
A reprodução sexuada proporciona a combinação de boas características dos indivíduos fazendo com que perpetuem com a

evolução das gerações. Observa-se que o vírus, ainda que se reproduzindo assexuadamente, tem elevada adaptação ao ambiente. Mas como isso é possível ? Isto se deve a alta taxa de mutação existente em sua reprodução. Conclui-se portanto que a mutação é importante na reprodução dos organismos e necessária como operador, no caso de baixa diversidade, quando na população, os cromossomos comportam-se como se fossem vírus, em reprodução assexuada.

Na mutação são realizadas mudanças aleatórias nos genes, ou seja: caminhadas aleatórias. Este operador secundário permite escapar de máximos locais e garantir acesso a qualquer solução do espaço.

Tradicionalmente, o operador mutação é aplicado junto ao de cruzamento, pois a mutação decorre de erro durante o cruzamento, com baixa probabilidade. Cada alelo, tem probabilidade  $p_m$  de sofrer mutação, mas salienta-se que este valor é muito baixo.

Além do método tradicional de mutação, Gonçalves et al (2005) sugere que uma parcela da população nova seja formada por indivíduos gerados pelo algoritmo original da primeira população. Segundo os autores a intenção é de prevenir convergências prematuras da população. A Figura 4.12 ilustra as estratégias de formação de novas populações.



FONTE: GONÇALVES ET AL (2005)

FIGURA 4.12 – EXEMPLO DE ESQUEMA ALTERNATIVO DE GERAÇÃO DE POPULAÇÕES

Torna-se importante ressaltar neste ponto que parte da estratégia de escolha do método de reprodução está vinculada à forma em como os

indivíduos são codificados, ou representados. Dentre os diversos trabalhos encontrados na literatura envolvendo os algoritmos genéticos, constata-se que muito do que os difere está nesta forma de representação das soluções. Muitos, inclusive, traçam comparativos entre estas formas de representação, avaliando performances. Outros, ainda avaliam performances de operadores genéticos, fortemente relacionados às representações genéticas adotadas. Para tanto, serão observadas algumas formas de representação para posteriormente discutir-se sobre alguns operadores aplicados aos AG's em JSSP's.



## 5. MÉTODOS APLICADOS AO PROBLEMA DE AGENDAMENTO

O presente trabalho visa obter soluções eficientes para os problemas de agendamento em *Job-shop* dinâmicos, realistas, voltados à produção sob demanda e focados a produção de pequenos lotes/lotos unitários. Tal eficiência concerne à aproximação dos resultados encontrados e do tempo gasto nesta atividade, bem como a qualidade dos resultados segundo restrições já elencadas de prazo, tempo de produção e etc. Dentro de um procedimento científico, parte-se por caracterizar o trabalho proposto, classificando-o quanto à natureza, abordagem do problema, ponto de vista dos objetivos e procedimentos técnicos.

### 5.1 CLASSIFICAÇÃO METODOLÓGICA DA PESQUISA

Esta pesquisa é aplicada, pois gera conhecimentos de abordagem prática e dirigidos à solução de problemas específicos. Quanto à forma de abordagem, é quantitativa, pois considera os resultados encontrados nas evoluções dos algoritmos, traçando comparativos, classificando-os na busca dos objetivos propostos. Quanto a estes objetivos, é exploratória, pois tem construídas hipóteses, testando e envolvendo levantamentos bibliográficos. No tocante aos procedimentos técnicos, são conduzidas diversas consultas, seja em artigos, livros ou demais materiais publicados sobre o assunto. É também pesquisa experimental, uma vez que algoritmos serão construídos, testados e avaliados.

Desta forma, o presente capítulo irá focar especificamente o método e as características envolvidas no agendamento e no AG aplicado ao problema.

Diversos autores, como Candido (1997) e Branco et al (2007) corroboram estruturas onde são necessários algoritmos de agendamento (normalmente atuantes na qualidade ativa, ou ativa sem atraso), além do algoritmo de busca (por ex: AG, AS, BT). Pode-se somar ainda, estratégias de busca local, na expectativa de melhora deste último. Tais estratégias podem estar “embutidas” em operadores (p.ex. o MSXF – *Multi-step crossover fusion* citado por Yamada (2003)) ou então constituir etapa isolada (refinamento dos resultados do algoritmo de busca), como sugere Candido (1997).

O presente trabalho aborda, de forma próxima a estas, a busca por agendamentos eficientes da classe de problema tratado, se balizado em

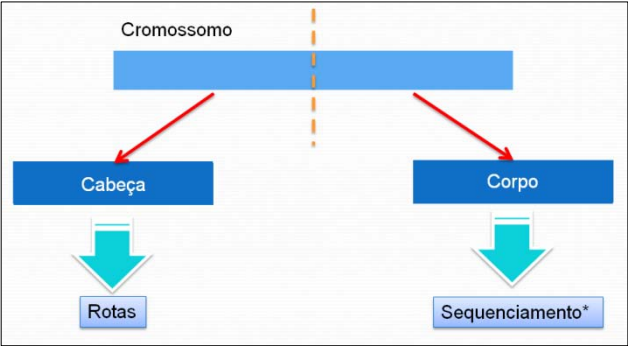
alguns pilares, a saber: detalhamento do problema *Job-Shop* e seara específica do problema a ser estudado; análise da geração de população inicial otimizada e sua influência nos AG's aplicados ao problema apresentado; geração e estudo da influência de aplicação de *schedulers* ativos e ativos sem atraso com uso de RHC's; estudos envolvendo RHC's no que diz respeito aos objetivos de menor tempo de processamento, cumprimento de datas devidas e/ou outro critério definido pelo usuário/sistema; ferramentas de simulação computacional; análise e ponderação dos resultados.

## 5.2 SOLUÇÃO DO JSSP PROPOSTO NO TRABALHO

Como mencionado, são estabelecidas algumas propostas para o desenvolvimento de uma estratégia inicial composta pelas seguintes etapas básicas: definição da codificação (formação do cromossomo), formação de população inicial, algoritmo genético, algoritmo de agendamento. Cada etapa será melhor apresentada nas sub-seções seguintes, contudo adianta-se que a avaliação dos indivíduos presentes na população utilizada pelos AGs é feita pelo algoritmo de agendamento. Este último opera integrado ao AG, decodificando as seqüências de máquinas para cada tarefa existente. A seguir tem-se a descrição de cada etapa.

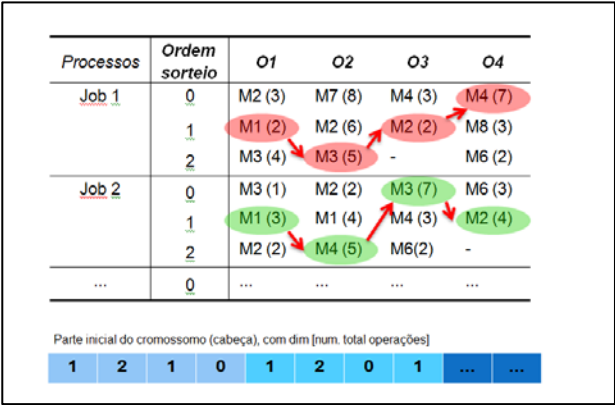
### 5.2.1 Estrutura do cromossomo proposto

Com o objetivo de realizar duas operações distintas, também o cromossomo tem duas partes bem definidas. A primeira, aqui denominada de “cabeça”, contém as informações pertinentes à alocação de recursos. Em outras palavras, é na “cabeça” que se registram as informações pertinentes as rotas entre as máquinas. A segunda parte irá atuar no seqüenciamento de operações. Como o seqüenciamento é guiado pela alocação, doravante esta outra parte do cromossomo será designada “corpo”. Tanto a “cabeça” quanto o “corpo” têm dimensões iguais a quantidade total de operações a serem realizadas. A figura 5.1 apresenta a estrutura geral do cromossomo.



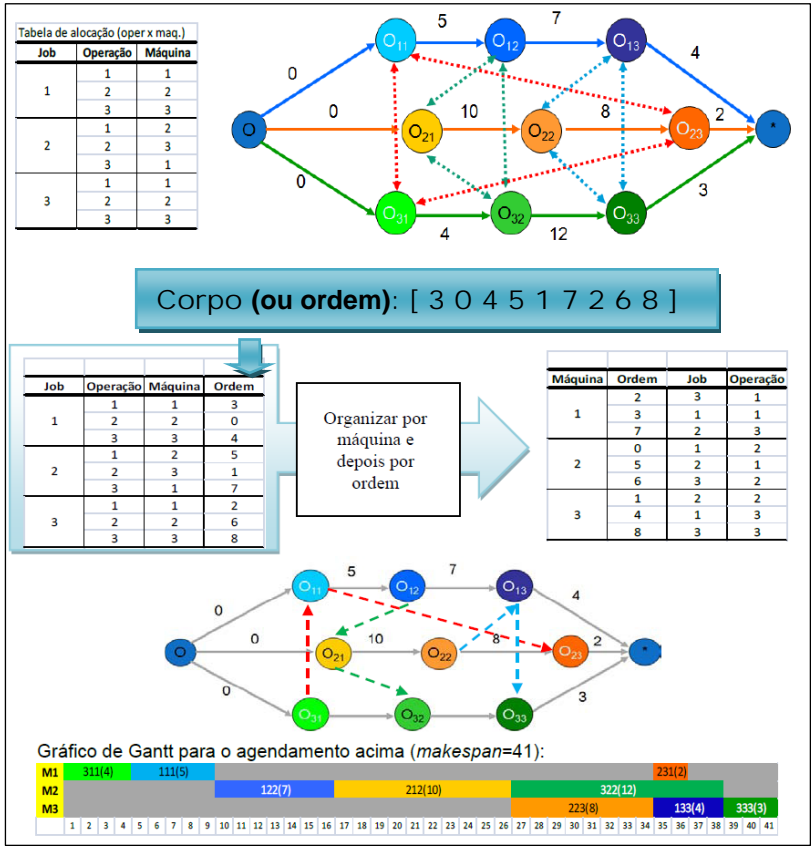
FONTE: PRÓPRIA  
FIGURA 5.1 – ESTRUTURA GERAL DO CROMOSSOMO: CABEÇA E CORPO

Na “cabeça”, para cada operação, o *locus* conterá o índice correspondente a máquina, na tabela original do cadastro feito pelo usuário, durante definição do problema. Na figura 5.2 tem-se a representação gráfica desta relação, onde observa-se a impossibilidade de formação de indivíduos inválidos, implicando não haver gastos com operações de harmonização cromossômica. Isto posto, constata-se a implementação de operadores como o de mutação, que poderá compor novas sugestões de recurso para processar uma dada operação sorteada, discutido posteriormente com mais relevância..



FONTE: PRÓPRIA  
FIGURA 5.2 – RELAÇÃO ENTRE TABELA ORIGINAL E CROMOSSOMO – FENÓTIPO x GENÓTIPO

Conhecido o total de operações, o “corpo” do cromossomo contém alelos inteiros, não repetidos, no intervalo  $a_i=[0,\text{total de operações}-1]$ . No processo de agendamento, o “corpo” indexa as operações para ordenação, definindo a hierarquia de processamento das operações, para cada máquina. Em outras palavras, consiste em converter arcos disjuntivos em arcos conjuntivos de um grafo de *job-shop*. A Figura 5.2 exemplifica esta relação.



FONTE: PRÓPRIA  
FIGURA 5.3 – RELAÇÃO GENÓTIPO X FENÓTIPO NO SEQUENCIAMENTO DE OPERAÇÕES

Um cuidado importante na definição desta etapa reside na aplicação dos operadores genéticos como o *crossover* e a mutação.

Como sua estrutura é simples e consiste em indexadores, está livre de gerar (ou tornar) indivíduos inactiváveis. Posteriormente, ao discutir-se a etapa de agendamento (avaliação dos indivíduos) e também dos operadores genéticos envolvidos no trabalho, maiores detalhes serão apresentados.

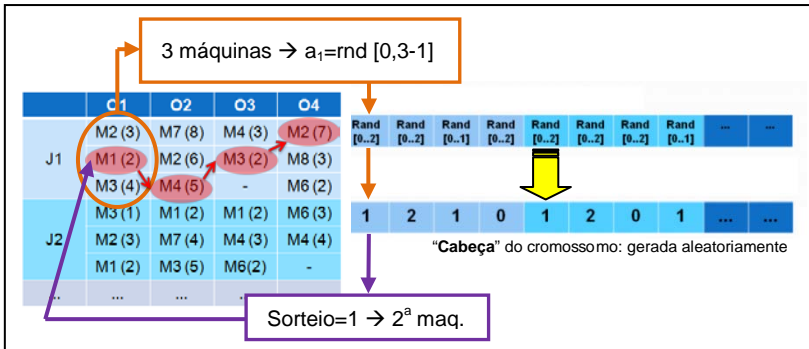
Neste ponto é interessante iniciar o delineamento das estratégias de trabalho pela formação da população inicial, contudo este conjunto de soluções faz parte dos algoritmos genéticos envolvidos. Assim sendo, parte-se por apresentá-lo primeiramente. O algoritmo genético proposto será um hibridismo de algumas técnicas associadas ao AG tradicional, onde inicialmente serão desenvolvidos estudos sobre seu funcionamento com população aleatória, com população gerada por critérios gulosos e população obtida com ambas às características. Com a geração de diversas soluções iniciais aliando características randômicas e gulosas, objetiva-se ganhar tempo computacional com pré-processamento no algoritmo.

### 5.2.2. Formação da população inicial

Com o cromossomo constituído de duas partes (cabeça, com a alocação de recursos e corpo, com o seqüenciamento das operações às máquinas alocadas), a criação de uma geração inicial irá incidir diretamente na criação de indivíduos aptos nesta estrutura.

Com o seqüenciamento dependente da escolha das rotas de máquinas, entende-se que a primeira parte do cromossomo deve ser construída antes da segunda. Assim, da tabela original, definida pelo problema estudado, são extraídas algumas informações importantes, como a quantidade total de operações e a quantidade de máquinas disponíveis para realizar cada uma das operações.

A construção da “cabeça” do cromossomo é feita pelo sorteio de números aleatórios no intervalo de  $[0, nm_{ij}-1]$ , sendo  $nm_{ij}$  o número de máquinas disponíveis a processar a operação  $j$  do processo  $i$ . Na figura 5.2 tem-se um exemplo de cromossomo e a alocação de recursos e na Figura 5.4 é exemplificada formação aleatória deste mesmo fragmento de cromossomo.



FONTE: PRÓPRIA

FIGURA 5.4 – EXEMPLO DE GERAÇÃO ALEATÓRIA DE “CABEÇA” DE CROMOSSOMO

No caso de formações gulosas SPT ou LPT é designado para o cromossomo o índice referente à máquina com menor ou maior tempo de processamento, respectivamente. Como exemplo, o primeiro alelo do cromossomo seria 1, se empregada a SPT ou 2, se empregada a LPT. Isto porque a máquina com menor tempo de processamento está associada ao índice 1 (M1, com 2 u.t.) e a com maior tempo de processamento está relacionada ao índice 2 (M3, com 4 u.t.).

Realizada a geração das alocações de recursos, parte-se para a construção do “corpo” do cromossomo. Neste fragmento do cromossomo é distribuído um conjunto de números inteiros, não repetidos e dispostos de forma aleatória. A forma de fazê-lo consiste em criar um vetor ordenado de 0 até  $nm_{ij}-1$  e posteriormente embaralhá-lo.

### 5.2.3. Função aptidão e avaliação dos indivíduos

A aptidão, ou *fitness*, de cada indivíduo é dada a partir da avaliação dos mesmos. Entendendo-se aptidão como a quantificação da adaptação do indivíduo, consiste na aplicação deste na função objetivo. Assim, após ser construída uma população, cada indivíduo é submetido à função objetivo para que se torne conhecida sua aptidão.

Como os AG's são focados na maximização (melhores indivíduos, os mais adaptados, etc.) e deseja-se os menores tempos de processamento, menores atrasos, etc., será preciso ajustar a avaliação para uma que seja inversamente proporcional, a fim de que se promova a maximização.

### 5.3.3.1. O agendamento como função de avaliação

Para cada indivíduo o procedimento de avaliação se repete. Considerando as rotas definidas no cromossomo, tem-se o conhecimento das máquinas a realizar cada operação. Da mesma forma, tem-se a sequência tecnológica de cada processo e a sequência de operações para cada máquina, esta também oriunda do cromossomo.

Ordenadas as operações para cada máquina, conforme o “corpo” do cromossomo, o agendamento ativo segue o seguinte algoritmo, que é uma modificação do proposto por Giffler e Thompson (1960).

#### ***Algoritmo G & T\_A-modificado***

##### ***Início***

**Passo 1:** Colocar a primeira operação agendável de cada tarefa (do horizonte de planejamento ativo) no conjunto de operações candidatas  $C = \{o_{il} \mid 1 \leq i \leq n\}$ ;

**Passo 2:** Escolher uma operação  $o'$  de  $C$ , com tempo de término mais cedo;

**Passo 3:** Determinar a máquina  $M'$  na qual  $o'$  deve ser processado e assim construir o conjunto  $G$  (denominado *conjunto conflito* de  $M'$ ), constituído de todas as operações de  $C$  a serem executadas em  $M'$ ;

**Passo 4:** Remover as operações que não iniciem antes de finalizar  $o'$ ,  $G = \{o_{ik} \in G \mid t_{ik} < t' + p'\}$ ;

**Passo 5:** Executar sub-algoritmo para selecionar uma operação  $o_{ik}^*$  de  $G$ ;

**Passo 6:** Remover  $o_{ik}^*$  de  $C$ ,  $C = C \setminus \{o_{ik}^*\}$

**Passo 7:** Inserir a operação  $o_{ik}^*$  no agendamento e calcular o tempo de início;

**Passo 8:** Inserir a operação sucessora de  $o_{ik}^*$  no conjunto  $C$  (caso exista);

**Passo 9:** Se  $C \neq \emptyset$ , ir para Passo 2, senão FIM.

##### ***Fim do algoritmo***

Para executar o Passo 5 é preciso executar a sub-algoritmo a seguir:

<b>Sub-algoritmo Escolher_oik*_de_G</b>
<b>Início</b>
<p>Criar CR: conjunto de operações a usar <math>M'</math>, não agendadas, com antecessoras agendadas;</p> <p>Se <math>G \cap Cr \neq \{\}</math> então</p> <p>Escolher operação de <math>G \cap Cr</math> com menor índice de ordem;</p> <p>Senão</p> <p>Criar PG: conjunto de processos pertencentes às operações de G;</p> <p>Criar CrG: sub-conjunto de Cr, com as operações <math>\in</math> aos <i>jobs</i> de PG;</p> <p>Se <math>CrG \neq \{\}</math> então</p> <p>Encontrar O: operação de CrG com maior índice de prioridade;</p> <p>Encontrar J*: processo que contenha a tarefa O;</p> <p>Encontrar O*: operação de G pertencente a J*;</p> <p>Senão</p> <p>Encontrar O*: operação de G com menor prioridade;</p> <p>Fim Se;</p> <p>Fim Se;</p>
<b>Fim do algoritmo</b>

Quando desejado operar com formações baseadas em agendamentos ativos - sem atraso, algumas alterações são propostas ao algoritmo GT (GIFFLER E THOMPSON, 1960; ALMEIDA, 1995):

<b>Algoritmo G&amp;T_A_ND-modificado</b>
<i>// nenhuma máquina fica livre se for possível iniciar nela algum processo:</i>
<b>Início</b>
<b>Passo 2:</b> Escolher uma operação $O'$ de C, com <u>tempo de início mais cedo</u> ;
<b>Step 4:</b> Remover <u>as operações que iniciam antes de <math>O'</math></u> ,
$G = \{o_{ik} \in G \mid t_{ik} = t'\};$
<b>Fim do algoritmo</b>



O objetivo da modificação é propor “uma espécie” de método de projeção sobre um conjunto de operações, ainda que não se disponha daquela indicada no sequenciamento prévio. O intuito é tentar conservar ao máximo a sequencia de operações definida no “corpo” do cromossomo para as máquinas, sem deixar de lado as regras de prioridade de despacho.

Se, para a máquina disputada para agendamento  $M'$ , a operação coincidir em ser a sequenciada pelo “corpo” do cromossomo e a mais prioritária pela regra de despacho, esta será também a operação  $O^*$ , ou seja, aquela operação candidata que foi eleita para agendamento.

Entretanto, como nem sempre o sequenciamento posto pelo “corpo” do cromossomo coincidirá com a sequencia tecnológica do *job-shop*, conflitos podem ocorrer.

Como há dois interesses em conflito, deve-se encontrar uma forma de considerar os dois sem que indivíduos inaptos sejam concebidos. A forma encontrada para promover a harmonização destas incompatibilidades é tentar encontrar em  $G$  uma operação que mais se aproxime daquela sugerida pelo “corpo” do cromossomo. Em outras palavras, para a máquina  $M'$ , procura-se agendar uma operação de  $G$  que pertença ao mesmo processo da operação mais prioritária do conjunto  $Cr$ .

Caso não existam elementos na intersecção de  $G$  com a operação indicada pelo sequenciamento do cromossomo para  $M'$ , então se opta pela operação de  $G$  menos crítica, ou seja, menos prioritária, relaxando assim tanto num quanto noutro critério de escolha.

#### 5.2.3.2. A aptidão relativa à função avaliação

Calculados todos os agendamentos, tem-se a menor e a maior resposta da função adaptação da população avaliada. Contudo, como já discutido em seções anteriores, indivíduos com menores tempos de finalização, ou com menores atrasos, enfim, com menores valores na função de avaliação são os melhores indivíduos. Observa-se o interesse na minimização da função aptidão. Este dado é importante, pois irá afetar diretamente a escolha da forma de aplicar o método de seleção.

O método da roleta, a exemplo, necessita adaptar a fórmula de *fitness*, uma vez que a forma de pensar é que “os mais aptos devem ser selecionados”.

No método escolhido, o de Mayerle, esta adaptação é mais simples, pois depende apenas da forma como promover a ordenação prévia que o método exige. Posteriormente basta gerar a variável

aleatória que indicará o indivíduo selecionado. Os procedimentos são melhor descritos nas seções que seguem.

#### 5.2.4. A seleção de indivíduos

O processo de seleção deve elencar os indivíduos que irão participar da reprodução. Neste modo, emprega os valores de *fitness* já calculados na etapa anterior para desenvolver as probabilidades de seleção.

O método de seleção adotado é o proposto por Mayerle (1994) que se baseia na escolha de indivíduos considerando distribuição de probabilidade inversamente proporcional ao *rank* dos indivíduos. Como visto no Capítulo 4, este método tem pressão seletiva inferior ao de Monte Carlo, contudo evita formação de superindivíduos.

##### 5.2.4.1 Estratégia de formação de população

O método de Mayerle escolhe indivíduos mediante, indiretamente, suas aptidões. Partindo de uma ordenação crescente inicial e de uma variável aleatória, ainda que apresente uma menor pressão seletiva que o método a roleta, indivíduos mais aptos terão mais chances de sucesso que os demais, o que leva, ainda que com leve tendência, a uma homogeneidade da população: filhos destes pais, ao longo das gerações, tenderão a dominar a população. Os elitismos (esta perpetuação dos melhores indivíduos em populações correntes) são discutidos como adequados para problemas como os desta tese (MATTFELD, 1996; GONÇALVES ET AL, 2005), mas exigem taxas mais elevadas de mutação para escapar de máximos locais. Salienta-se que estas taxas, ao se elevarem, aumentam junto o custo computacional.

A estratégia adotada para seleção consiste em conciliar a robustez do método de Mayerle com a diversidade proporcionada por indivíduos gerados aleatoriamente e a “dopagem” proporcionada pela geração gulosa empregando regras de despacho. Principalmente os de formação aleatória, em muito maior parcela de inserção, sempre permitirão dispor de diversidade de indivíduos na geração corrente. Também a mutação irá proporcionar alterações cromossômicas e aliar-se nesta ampliação dos espaços de busca, mas sem demandar de grandes taxas.

##### 5.2.4.2 O método de seleção de Mayerle

Conforme visto no Capítulo 4, a implementação do método de

Mayerle é bastante simples. Conhecidos os valores da função adaptação de cada indivíduo, elabora-se uma lista ordenada destes, segundo critério crescente de *fitness*. A variável aleatória dada pela Eq. 8 tem a função de indicar a posição do indivíduo selecionado na lista em questão. O indivíduo indicado fará par com outro para então ser submetido ao operador de cruzamento e mutação.

Este método suprime a presença de superindivíduos, uma vez que opera indiretamente sobre os valores da função de adaptação dos indivíduos.

Uma grande parte da população é selecionada para cruzamento e outra, pequena, sofrerá ação de procedimentos como: clonagem (elitismo), formação aleatória e gulosa.

### 5.2.5. Os operadores genéticos

Empregam-se três operadores na estratégia evolutiva proposta: cruzamento, mutação e clonagem. As particularidades de cada um são agora discutidas.

#### 5.2.5.1 Operador de cruzamento (*crossover*)

O operador de recombinação (cruzamento ou *crossover*) é o principal operador do algoritmo genético, e como já apresentado, visa perpetuar as características dos indivíduos mais aptos através da troca de informações destes pais, repassando-as aos indivíduos filhos.

O operador de cruzamento aplicado na cabeça difere do operador aplicado no corpo do cromossomo. Na primeira parte são transportadas as chaves randômicas (os alelos são números aleatórios que indexam a tabela original de máquinas disponíveis para processar cada operação) para os filhos.

O operador aplicado é o crossover uniforme, que se baseia na construção de uma máscara binária, um vetor formado por dígitos “0” e “1”, atribuídos conforme distribuição uniforme. A dimensão desta máscara é a mesma da “cabeça” do cromossomo.

Durante o processo de cruzamento desta parte do cromossomo, tem-se que um alelo igual a “zero” implica ao filho o gene do pai 1. Se o alelo da máscara for “1”, tem-se que o doador do gene é o pai 2. Para gerar outro filho, adota-se a inversão da máscara e a submissão dos mesmos pais, novamente, na função de cruzamento. Para cada par de filhos gerados a máscara é refeita, visando explorar melhor as combinações de genes dos indivíduos.

Não há necessidade de harmonizações pois o operador escolhido atua sobre chaves aleatórias compatíveis, haja visto dizerem respeito ao mesmo campo indexador na matriz original de máquinas e operações.

Após realizar esta etapa, os indivíduos são submetidos à segunda fase do cruzamento, agora empregando operador baseado em ordem. Isto é necessário pois se deseja manter com maior fidelidade possível a sequência proposta pelos pais. O operador aplicado é o PPX (*precedence preservative crossover*) que atua da seguinte forma sobre os cromossomos (no caso, no “corpo” deste): utilizando a mesma máscara envolvida na etapa anterior. Herdando do pai 1 todos os genes situados nos respectivos loci da máscara com alelo “zero”, parte-se por completar a sequência baseando-se no pai 2. Se o alelo obtido do pai 2 já existir em algum locus do filho 1, busca-se o próximo, até que seja possível completar aquela lacuna. O procedimento é repetido para todas as lacunas do filho 1.

Da mesma forma, constrói-se o filho 2, contudo para este, um “zero” na máscara implica em adquirir o gene do pai 2; um “um” implica em um gene do pai 1.

Como observado no capítulo 2, este operador foi aplicado tomando-se como base as observações de Gonçalves et al (2005) que tais operadores surtem bons efeitos quando aplicados a agendamentos, no lugar dos tradicionais um ou dois pontos de corte.

#### 5.2.5.2 Operador de mutação

O operador de mutação é aplicado em conjunto aos operadores de cruzamento, com probabilidade definida pelo usuário. Evidentemente que para cada parte do cromossomo terá um procedimento associado diferente na mudança dos genes do cromossomo, e assim dois valores de probabilidade de atuação serão definidos, um para cada fase de cruzamento.

Durante o processo de cruzamento da “cabeça” do cromossomo, o operador mutação irá agir alterando o valor sorteado por um novo, também sorteado aleatoriamente. Este método denomina-se denominado *mutação aleatória*. Sob o aspecto do fenótipo, uma nova máquina poderá ser designada àquela operação, implicando em uma alteração de rota.

No processo de cruzamento do “corpo” do cromossomo, a mutação irá atuar fazendo uma troca de genes, entre dois pontos sorteados aleatoriamente. Denominada *mutação por troca (swap)*, ela forçará uma alteração na sequência de execução de tarefas. Esta

alteração é sutil pois não atua especificamente sobre o seqüenciamento de uma determinada máquina, mas globalmente em todos os seqüenciamentos. Com a probabilidade de não surtir efeitos significativos durante o processo, contudo ganha no aspecto de atuar sobre todas as seqüências.

#### 5.2.6. Os parâmetros do AG

Tocante aos parâmetros envolvidos no AG, além da escolha dos métodos aplicados em cada uma das etapas do AG, alguns parâmetros devem ser definidos pois incidem diretamente na performance total da busca da melhor rota e melhor agendamento, conforme os objetivos estipulados para o problema.

Os parâmetros que denotam maior influência são: tamanho da população; o número de gerações; probabilidade de cruzamento e probabilidade de mutação. Outros, secundários ao sistema, são: formação de população inicial; tipo de agendamento; regra heurística aplicada no agendamento.

Os critérios de definição destes parâmetros serão, agora, melhor discutidos:

##### 5.2.6.1 Tamanho da população

O número de indivíduos afeta o desempenho global do AG, pois se pequenas, podem não cobrir todo o espaço de busca e, se forem grandes demais, podem exigir um esforço computacional excessivo para solucionar o problema proposto (REEVES,1995).

Se a avaliação do cromossomo é lenta, é conveniente o uso de pequenas populações. De Jong apud Mitchell (1996) relatam aplicações entre 50 e 100 indivíduos. Gonçalves et al. (2005) sugere populações com o número de indivíduos igual ao dobro no número de operações existentes.

Estabelecendo uma relação apropriada entre a quantidade total de rotas possíveis é possível garantir uma proporção fixa para aplicar os algoritmos genético e de agendamento.

##### 5.2.6.2. Critérios de parada do algoritmo

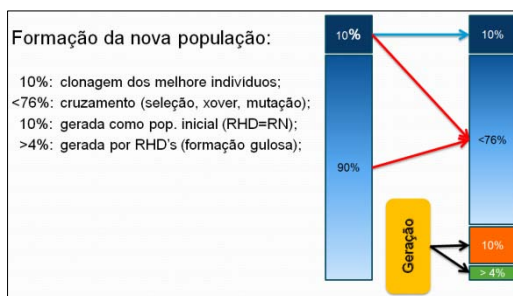
Uma vez definidas as prioridades de atuação, alguns parâmetros podem servir de balizamento para interromper um processo iterativo de busca de solução para o problema estudado. Salienta-se que muitas

vezes emprega-se como critério de parada um *set point* máximo para erro em relação ao ótimo, se conhecido. Quando o erro máximo não se enquadra na faixa desejada, um menos prioritário poderá atuar. Geralmente, o número máximo de operações enquadra-se neste plano. Isso significa que se não atingido o erro relativo desejado, ao atingir-se uma determinada quantidade de gerações, o algoritmo é interrompido.

Contudo, para problemas realistas, outros critérios podem estar relacionados, como por exemplo os tempos de finalização de cada operação e o prazo máximo designado a cada uma delas. Se a folga for negativa ou insuficiente, o sistema continua atuando até que um resultado satisfatório seja alcançado. Se não atingido, o algoritmo é interrompido.

### 5.2.6.3. Taxas de cruzamento

As taxas de cruzamento definem se será ou não feito o cruzamento entre dois indivíduos selecionados. Conforme literatura, esta taxa é entre 0,6 e 0,7. Estes valores indicam que, com uma probabilidade  $0,6 \leq p_c \leq 0,7$  um indivíduo poderá ser designado a realizar cruzamento após ter sido submetido à seleção. Se os pais não cruzarem, serão copiados para a população nova (clones). Sabe-se que taxas muito baixas tornam o desempenho do algoritmo mais lento, visto menor potencial de exploração do espaço de busca e maior elitismo. Tomando-se como base a estratégia de ação definida para o método proposto, esta probabilidade existirá, contudo indiretamente, haja visto designar-se indivíduos para clonagem e para formação aleatória. Os demais, certamente, serão conduzidos para cruzamento com probabilidade de 100%. Em geral, a probabilidade  $p_c$  seria de aproximadamente 70%. Na Figura 5.5 é ilustrada esta estratégia de formação.



FONTE: PRÓPRIA

FIGURA 5.5 - ESTRATÉGIA PARA A FORMAÇÃO DAS POPULAÇÕES

#### 5.2.6.4. Taxas de mutação

Atuante em duas fases, a mutação tem também duas probabilidades de ocorrência durante os métodos de cruzamento. De maneira geral, serve para indicar se os genes sofrerão ou não alteração, e assim, com taxas muito baixas os genes podem ter seus valores estagnados no tempo. Com taxas muito elevadas, ao contrário, as buscas tornam-se essencialmente aleatórias, onde praticamente são desprezadas as informações históricas / herdadas dos indivíduos anteriores. Na literatura, os valores das probabilidades de mutação genética são baixos, na ordem de 0,1 a 5%.

Com base na estratégia apresentada por Gonçalves et al (2005) também é desejada uma taxa reduzida, justificada pela característica da estratégia, onde já são inseridos indivíduos gerados aleatoriamente, a cada nova formação de população, garantindo quebras na heterogenia da população. Observa-se que  $p_m$  elevadas podem garantir diversidade para populações tendenciosamente homogêneas, contudo elevam o custo computacional e reforçam as caminhadas aleatórias. Os valores a serem praticados para mutação são da ordem de até 5 % na fase de construção de rotas e também na de seqüenciamento de operações, dada a pouca força do operador de mutação sobre uma determinada sequência, de uma determinada máquina.

### 5.3 REGRA HEURÍSTICA COMBINADA – RHC

No tocante aos interesses do trabalho, é importante considerar os prazos devidos na execução dos processos, dadas às características de demanda orientadas a encomenda/pedidos que os processos estão sujeitos, mas sem esquecer-se das condições pertinentes a tempos de fluxo nos processos, o que conduz a regras combinadas.

Com base nas informações colhidas na literatura, utilizar-se de regras mais simples para efeitos de combinação poderão surtir efeitos melhores quanto ao tempo computacional. Assim sendo, alia-se as regras  $SPT_q$  (menor tempo que falta para finalizar um processo), na Eq. 5.1 e a  $S/RPT$ , em Eq. 5.2, motivados pelos bons resultados da primeira em uma grande variedade de trabalhos e no caso da segunda, por considerar o tempo que falta para finalizar os processos em andamento.

$$iSPT_q = \sum_{q=j}^{mi} piq \quad (5.1)$$

$$iSPRT = \frac{d_i - t - \sum_{q=j}^{mi} piq}{\sum_{q=j}^{mi} piq} = \frac{d_i - t - iSPT_q}{iSPT_q} \quad (5.2)$$

onde:

$\sum_{q=j}^{mi} piq = iSPT_q =$  tempo de processamento que falta par ao término do job<sub>i</sub>;

Os dois índices são inversamente proporcionais ao grau de prioridade que indicam a operação, ou seja, maior prioridade para valores menores e menor prioridade para valores maiores, logo o algoritmo é construído para priorizar as operações com menores índices.

O índice iRHC é dado pelas Eq. 5.3 e 5.4, abaixo:

$$iRHC = iSPT_q \cdot iSPRT = iSPT_q \cdot \frac{d_i - t - iSPT_q}{iSPT_q} \quad (5.3)$$

$$iRHC = d_i - t - iSPT_q \quad (5.4)$$



## 6. APLICAÇÃO DA PROPOSTA DE AGENDAMENTO E ANÁLISE DOS RESULTADOS: TESTES PARA CASOS CLÁSSICOS E REALISTAS

Com o intuito de avaliar o comportamento geral dos algoritmos genético e de agendamento, definiram-se cinco seqüências de testes, elucidadas nas demais seções deste capítulo e aqui citadas: **comportamento diante de casos clássicos** (sem rotas alternativas – desempenho do agendador e observação das heurísticas); **desempenho diante do aumento de rotas alternativas** (avaliação em problema da literatura antes de depois de expandida a flexibilidade de rotas; **desempenho do método proposto diante de** casos da literatura envolvendo **perturbações** (desempenho simulado diante de casos NDDJSSP); **observância isolada de comportamento diante de *setup* dependente** (capacidade de escolha de uma tarefa em detrimento de outra para escalonamento, conforme tempo dependente de sequencia e disponibilidade de seleção); **observação quando emulada a quebra de equipamentos/parada de máquinas** do algoritmo de tomar decisão diante de fato inesperado). Os problemas e a tabulação dos resultados encontram-se dispostos nos Apêndices.

Como o cromossomo é organizado em duas partes, também o algoritmo irá atuar diferentemente em cada uma delas. Como já apresentado, na etapa “corpo” são tecidos os agendamentos com base nas rotas traçadas pela parte “cabeça”. Desta forma, em um primeiro momento, propõe-se testes para demonstrar e avaliar o comportamento da parte “corpo”, para posteriormente poder avaliar o conjunto “cabeça-corpo” ou roteamento-agendamento. Para tanto, uma forma encontrada de avaliar o desempenho da etapa de agendamento é com problemas clássicos da literatura, haja visto que nestes casos, as rotas são únicas.

Da literatura, foram escolhidos os problemas ABZ6, FT10, LA02 e LA21. Bastante conhecidos, foram obtidos na biblioteca *OR-Library* de Beasley (2007), permitem avaliar o desempenho para agendamentos sem rotas alternativas e resultados ótimos conhecidos.

Em um segundo momento, depois de já conhecido o desempenho dos algoritmos na fase “corpo”, tem-se que avaliar o comportamento da parte “cabeça”. Para tanto, serão aplicados problemas como os de Chan (2003) e Kumar et al (2003), que disponibilizam rotas alternativas para processos. Nesta fase são observadas as capacidades de operação com rotas alternativas, avaliando-se simultaneamente o comportamento da conjuntura proposta junto a heurísticas, taxas de probabilidade de

mutação pré-selecionadas, tamanho de população e tempos computacionais gastos para atingir resultados satisfatórios.

Isoladamente são desenvolvidos dois conjuntos de ensaios para avaliar o comportamento diante de quebras de máquinas e também *setup* dependente de sequência.

A seguir discorre-se sobre os métodos aplicados para os testes propostos, e posteriormente reúnem-se as principais discussões sobre os resultados encontrados.

## 6.1 TESTES DA ETAPA AGENDAMENTO – CASOS CLÁSSICOS DE JSSP

### 6.1.1 Métodos adotados para os testes

Para os testes de agendamento, visando avaliar o comportamento de diferentes heurísticas junto ao algoritmo GT modificado, selecionou-se da literatura quatro problemas conhecidos: ABZ6 de Adams, Balas e Zawack (1988); LA02 e o LA21, de Lawrence (1984); FT10 (MT10), de Fisher e Thompson (1963).

De acordo com Vaessens et al (1986) e de Gonçalves et al (2005), os problemas LA21 e FT10 são mais difíceis de solucionar, ao contrário dos ABZ6 e do LA02.

Para executar os testes é preciso definir as estratégias de aplicação: parâmetros do AG, regras heurísticas, quantidade de dados gerados por combinação e tipo de agendamento.

Dadas as características dos AGs, os parâmetros mais importantes a serem definidos são: tamanho e formação da população, taxa de mutação e função objetivo.

Para a formação da população, será seguida a estratégia apresentada no Capítulo 5, semelhante à proposta por Gonçalves et al (2005). De tamanho fixo e previamente ordenado conforme a função objetivo escolhida, tem clonados para a população seguinte, os primeiros indivíduos em número definido previamente. Esta clonagem é responsável pelo elitismo no AG. Como forma de compensar este elitismo, outra parte da população é criada aleatoriamente como na primeira população, embora em baixa proporção. Também como forma de garantir uma convergência mais rápida, mas não prematura, outra pequena parcela de indivíduos menos aptos é substituída por indivíduos gerados de forma gulosa, conforme regras heurísticas SPT, SPTq, S/PRT e RHC.

As inserções aleatórias e gulosas (para cada regra heurística)

seguem as proporções de 10%, 1%, 1%, 1%, 1% respectivamente. Arbitra-se para clonagem, a proporção de até 10%. Salienta-se que o operador de *crossover* atua com taxa de 100%, ou seja, todos os indivíduos são aptos a seleção e cruzamento.

Para o operador mutação, adota-se duas taxas experimentais: 1 e 5%. Acredita-se que, com as combinações geradas destes parâmetros será possível observar o comportamento evolutivo da população inicial até ser alcançado o critério de parada.

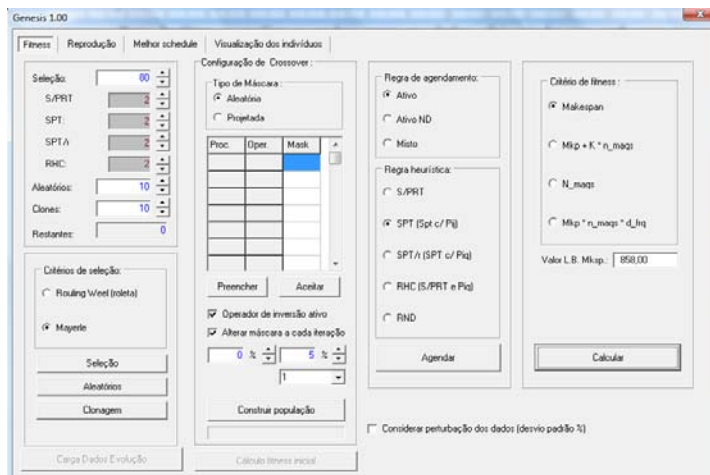
Os procedimentos irão gerar 11 soluções para cada combinação de parâmetros do problema e descartar, se for o caso, a menos próxima da média.

Quanto aos procedimentos de agendamento, baseando-se no algoritmo GT, uma modificação irá permitir gerar, além das formações ativas, também ativas sem atraso (A ou A-ND). Também será observado o desempenho frente o uso de populações de tamanhos diferentes (equivalente a 50 e 100% do total de operações a ser executado. P. ex.:  $FT_{10} = 10 \times 10 = 100$  operações  $\rightarrow$  população = 100 indivíduos, ou 50% = 50 indivíduos ). No total, para cada problema são necessárias 400 execuções do programa. Como experimentalmente observou-se que para 2000 iterações já são encontrados bons resultados, fixou-se este valor como critério de parada para o algoritmo.

Para a escolha da função objetivo, fixou-se o *makespan*, visando compatibilizar os resultados com os da literatura, para efeito comparativo de desempenho.

Para facilitar a análise e gerar os dados, foi criado um *software* em linguagem *object Pascal* no Delphi 2009 ®. Com o nome de Genesis já na versão 2.32, o referido programa permite realizar as iterações com base nos parâmetros de formação de população, taxa de mutação e critério de parada baseado em numero máximo de gerações, partindo de uma população inicial construída. Também é gerado, ao final da análise, relatório com os dados da melhor solução, arquivo Excel ® com os dados de evolução, tempo e valor da função objetivo. Também é possível exportar gráficos de evolução em função objetivo escolhida, *Gantt* de qualquer indivíduo e tabelas de operação x tempo inicio-termino. Nos casos clássicos, para efeito comparativo, a função objetivo é o *makespan*.

A seguir, em uma tela do programa Genesis 2.32, é possível observar os dados solicitados para configurar a busca por soluções dos JSSP.



FONTE: PRÓPRIA

FIGURA 6.1 – TELA DO PROGRAMA GENESIS 2.32: CONFIGURAÇÃO PARA CÁLCULO

No estudo de Vaessens et al (1986) é citado o AG proposto por Dondorf e Pesch (1995) com características semelhantes ao empregado nesta tese, ou seja, um AG baseado em regras de prioridade e algoritmo GT. Os resultados, dispostos na seção 6.2 juntamente com os valores ótimos, permitem observar o desempenho da estratégia proposta.

A plataforma dos testes baseia-se em processador Core2 Duo E7400 com 2GB de RAM e Windows XP SP3. Os resultados, organizados em planilhas, estão nos Apêndices.

### 6.1.2 Resultados dos testes de agendamento clássico

Nesta seção são discutidos os resultados computados nos ensaios dos casos clássicos.

Os gráficos e tabelas geradas estão contidos nos Apêndices B a E.

Torna-se importante elucidar, antes de qualquer análise, que é desejado obter bons resultados, mas que não é objetivo atingir resultados ótimos, haja visto que o AG não apresenta qualquer técnica de busca local, o que segundo diversos autores, é o que torna o AG mais rápido e eficiente na busca por atingir resultados ótimos. Interesse é, portanto, conseguir agendamentos bons em tempo computacional baixo.

A seguir serão tecidas algumas breves considerações sobre cada problema estudado, a fim de posteriormente tecer as considerações

finais sobre os mesmos, buscando elucidar pontos que auxiliem na sintonia do AG para uso em estruturas mais complexas, como é o caso dos problemas realistas com múltiplas rotas e proposição de alguns ajustes também na heurística RHC, aplicada nos ensaios.

### 6.1.2.1. Problema LA02

Nas considerações relacionadas ao tempo, em média as vantagens estão mais associadas com esquemas envolvendo menores populações e taxas maiores de mutação. Aquelas empregando 5% de probabilidade estabeleceram tempos menores que 1 min., e em contrapartida, aquelas envolvendo 1% ficaram entre tempos de 1 min. até 1min e 30 s.

Outra característica importante neste *ranking* está relacionada aos erros médios, ficando entre 4 e 6%.

A heurística RHC lidera as melhores posições, embora a SPTi totalize maior percentual de frequência entre as cinco melhores posições de tempo mínimo, consumindo 2 min. em média.

Ambas as probabilidades de mutação (PM) parecem estar bem ajustadas, embora das 5 melhores posições, 60% estejam ocupadas com esquemas envolvendo PMs de 5%.

Já tocante ao erro, é notória a predominância da heurística RND como a mais hábil nos agendamentos de menores erros, posicionando-se com 60% das 5 melhores posições no *ranking* de menor erro médio e também menor erro mínimo encontrado.

	T med	E% med
RND_50_ND_5	00:33,342	3,57%
RHC_50_ND_5	00:23,712	5,71%
RND_50_ND_1	01:07,590	3,30%
SPTi_50_ND_1	00:56,350	4,09%
RHC_50_ND_1	00:45,550	5,07%
SPTi_100_ND_1	01:09,487	3,36%
SPRT_50_ND_5	00:57,375	4,82%
SPRT_50_ND_1	01:18,300	4,52%
RHC_100_ND_5	01:28,230	4,49%
SPTq_50_ND_1	01:31,525	4,55%

(a)

	T med	E% med
SPRT_50_ND_5	00:27,896	3,05%
SPRT_100_ND_5	00:53,491	2,97%
SPRT_100_ND_1	00:55,987	2,97%
RND_50_ND_5	01:24,677	2,97%
SPRT_50_ND_1	01:43,404	2,97%
RND_50_ND_1	01:45,775	2,97%
RND_100_ND_5	02:23,024	3,05%
RHC_100_ND_1	02:49,951	3,05%
RND_100_ND_1	03:13,887	3,05%
SPTi_100_ND_1	02:53,008	3,44%

(b)

QUADRO 6.1 – CLASSIFICAÇÃO DOS 10 MELHORES (A) TEMPOS MÉDIOS E (B) ERROS MÉDIOS DO PROBLEMA LA02: ORDENAÇÃO BASEADA NO BALANÇO TEMPO X ERRO.

Novamente, priorizando-se o erro relativo, não há diferenças

entre optar por PMs baixas ou altas, muito embora no tocante aos erros relativos mínimos a de 1% detenha praticamente a totalidade das 5 melhores soluções.

Em resumo, é possível avaliar a proximidade dos dados, o que deixa o foco não mais para erro relativo ao ótimo, mas sim para uma relação mais íntima deste parâmetro com o tempo computacional gasto em solucionar adequadamente o problema. Assim, sutilmente despontam heurísticas como a SPTi, muito embora predominem fortemente as heurísticas RND e RHC, tanto para dados médios como mínimos. Ainda inseridos na proposta de encontrar uma conjuntura eficiente, que conjure tempo computacional e erro relativo baixos, despontam as formações RND e RHC, aliadas com populações menores e taxas que podem ser tanto de 1 como 5%, dado que não mostraram soluções muito distantes nos testes feitos.

### 6.1.2.2. Problema ABZ6

Como no caso do LA02, também é um problema de dimensões e complexidade menores que os demais, apresentando desta forma um comportamento similar, embora com particularidades relativas à forma como apresenta suas operações e os seus tempos.

Aparentemente, ainda tocando no quesito complexidade, dada a forma como bons resultados foram encontrados e a frequência em que ocorreram, para diversas combinações de parâmetros, são, portanto o problema menos complexo dos que foram escolhidos.

	T med	E% med
SPTi_50_ND_5	00:00,378	3,90%
SPTi_50_ND_1	00:00,593	3,90%
SPRT_50_ND_5	00:27,896	3,05%
RHC_50_ND_5	00:30,608	3,82%
SPRT_100_ND_5	00:53,491	2,97%
SPRT_100_ND_1	00:55,987	2,97%
SPTq_50_ND_5	00:44,539	3,84%
SPTq_50_ND_1	01:04,541	3,73%
RND_50_ND_5	01:24,677	2,97%
RHC_50_ND_1	01:19,264	3,52%

(a)

	T med	E% med
RND_50_ND_5	00:33,342	3,57%
RND_50_ND_1	01:07,590	3,30%
SPTi_50_ND_1	00:56,350	4,09%
SPTi_100_ND_1	01:09,487	3,36%
SPTi_50_ND_5	01:48,959	3,73%
SPTi_100_ND_5	02:13,561	3,36%
RND_100_ND_1	02:58,667	2,81%
RND_100_ND_5	02:44,860	3,18%
SPTq_100_ND_5	02:18,720	4,15%
SPTq_100_ND_1	03:35,297	4,12%

(b)

QUADRO 6.2 – CLASSIFICAÇÃO DOS 10 MELHORES (A) TEMPOS MÉDIOS E (B) ERROS MÉDIOS DO PROBLEMA ABZ6: ORDENAÇÃO BASEADA NO BALANÇO TEMPO X ERRO

Em linhas gerais, em média foram despendidos tempos da ordem de 30s para encontrar soluções com erros médios menores de 3%. Para tanto, no *ranking* das 10 melhores soluções, apesar do equilíbrio existente entre as heurísticas, salienta-se a leve predominância da SPTi, mais presente no topo da lista. Também a RHC merece destaque, embora com menor expressão que a anterior.

Tocante às formações de população, aquelas empregando populações menores foram mais freqüentes entre as 5 primeiras posições dos *rankings*, principalmente quando organizados sob o critério de balaço tempo x erro relativo. Desta forma houve destaque para as formações com heurísticas SPTi, RHC e SPRT.

Conjugando melhor tempo de processamento e bons resultados (baixos erros relativos ao ótimo), a formação de baixa população, com heurística SPTi e taxa de 5 ou 1% é destaque nesta formação.

#### 6.1.2.3. Problema LA21

Tido como o problema mais complexo do roll selecionado para testes, o LA21 teve, para as 10 melhores soluções encontradas no quesito tempo médio de processamento, uma distribuição bastante igualitária para todas as heurísticas. Embora a SPTq tenha liderado, os tempos estão todos muito próximos, oscilando entre 4 e 6 min. para as 5 melhores soluções e indo até 12 min. quando observada a décima posição.

Nas estratégias envolvendo, inicialmente tempos de processamento e, posteriormente o erro médio, tem-se o domínio das pequenas populações junto às melhores soluções encontradas.

	T med	E% med
SPTi_50_ND_5	00:35,914	7,05%
SPTi_100_ND_1	01:43,327	6,30%
SPTi_50_ND_1	01:57,180	6,32%
RHC_50_ND_5	01:58,026	6,99%
RHC_50_ND_1	02:05,665	6,92%
RND_100_ND_1	03:17,916	5,81%
SPTi_100_ND_5	03:56,858	6,04%
RHC_100_ND_1	03:37,271	6,73%
RND_50_ND_5	03:38,589	6,92%
RHC_100_ND_5	04:35,202	6,82%

(a)

	T med	E% med
SPTi_100_ND_1	01:43,327	6,30%
SPTi_50_ND_1	01:57,180	6,32%
RHC_50_ND_5	01:58,026	6,99%
RHC_50_ND_1	02:05,665	6,92%
RND_100_ND_1	03:17,916	5,81%
SPTi_100_ND_5	03:56,858	6,04%
RHC_100_ND_1	03:37,271	6,73%
RND_50_ND_5	03:38,589	6,92%
RHC_100_ND_5	04:35,202	6,82%
RND_100_ND_5	05:39,659	6,45%

(b)

QUADRO 6.3 – CLASSIFICAÇÃO DOS 10 MELHORES (A) TEMPOS MÉDIOS E (B) ERROS MÉDIOS DO PROBLEMA FT10: ORDENAÇÃO BASEADA NO BALANÇO TEMPO X ERRO

Quanto às PMs, não houve despontar expressivo. A mais elevada, contudo apresentou sutil vantagem, talvez por estar aliada com as pequenas populações, em esquemas que conduziram aos bons resultados observados.

Em outra situação, ao se observar os dados segundo o erro médio, estando entre 7 e 10%, vêm por indicar que a heurística RND está mais apta a proporcionar bons resultados, todos bem próximos, mas que consomem elevado tempo computacional para tal feito ( de 6 a 24 min., em media – tempos menores para populações menores e tempos maiores para populações maiores). Em segundo plano, também as formações SPTi despontam como boas estratégias de solucionar o problema de forma eficiente quanto a erro médio.

Com os dados organizados pela relação tempo médio de processamento e erro médio, tem-se que as formações RND e a SPRT são as mais balanceadas, com erros médios em torno de 7,3 e 9,37% para os tempos aproximados de 6 e 8 min., respectivamente.

Em linhas gerais, é evidente a participação da formação RND, menor população e taxa de 5% (PM), sendo observada em geral, entre as 3 primeiras posições dos *rankings*. As formações com populações maiores, ainda que operando com baixas taxas de probabilidade de mutação e proporcionando o melhor resultado (erro mínimo), consumiu um tempo computacional 12 min. a mais ( e cerca de 20 vezes em módulo) que o segundo valor de erro, distando insignificantes 0,67% ( de 5,83 para 6,5%). A terceira posição consumiu 2 min. e 27 seg. atingindo 6,7% de erro, o que demonstra que embora em segunda posição, a conjuntura RND e pequenas populações têm forte correlação com os melhores resultados. Como reforço desta análise, considerando dados médios, torna-se interessante operar com PM igual a 5%, onde, para erros médios próximos, o tempo médio, é a metade que com 1%.

Neste problema, observado que as formações SPTi e SPTq foram pouco expressivas, também o foi a RHC, que combina linearmente as duas anteriores.

#### 6.1.2.4. Problema FT10

Predominando as heurísticas SPTi, RHC e RND, é notória a participação da SPTi nas melhores soluções tempo x erro médio. Da mesma forma, tem-se o predomínio de populações menores na composição das estratégias envolvidas nos melhores resultados.

Quando se discute os tempos de processamento, as formações SPTi também despontam como 1,7 a 3,4 vezes mais rápidas que a RND



e 5 vezes mais rápida que a RHC, embora estas operando com populações maiores e erros praticamente iguais aos seus.

Na busca considerando o erro em relação ao valor ótimo, observa-se o predomínio de populações maiores com taxas baixas de PM. Com 60% das 10 melhores colocações, ocupam 4 das 5 primeiras posições. A outra posição diz respeito à heurística SPTi, com população pequena e taxa PM baixa, em 1%.

Priorizando as análises em função do tempo médio, as formações com pequenas populações e taxas PM elevadas (5%) imprimiram bons resultados quando aliadas a heurísticas como a SPTi ou a RHC. Como a RHC emprega a SPRT e a SPTi, não teve participação mais expressiva devido a primeira não ter despontado nas classificações.

Quando priorizado o erro médio, também é consenso que as formações SPTi e RHC despontam como as presentes nas melhores elações erro % x tempo, sendo iminente a predominância de pequenas populações ( baixas taxas de PM tendem a contribuir para menores tempos computacionais, mas no caso, foram tênues diferenças).

	T med	E% med		T med	E% med
RND_80_ND_5	06:00,899	7,28%	SPTi_80_ND_5	03:55,129	10,55%
SPRT_160_ND_1	07:58,764	9,54%	RND_80_ND_5	06:00,899	7,28%
RND_80_ND_1	12:42,130	7,82%	SPTq_80_ND_1	03:42,105	12,73%
SPTq_160_ND_5	10:47,182	9,39%	RHC_80_ND_1	04:47,656	11,32%
RND_160_ND_5	16:09,744	7,53%	SPTq_80_ND_5	05:19,787	11,11%
SPTi_80_ND_1	12:19,668	10,19%	RHC_80_ND_5	06:12,307	10,71%
RHC_160_ND_5	13:27,977	9,37%	SPRT_160_ND_1	07:58,764	9,54%
RND_160_ND_1	24:28,807	7,11%	SPTq_160_ND_5	10:47,182	9,39%
SPTi_160_ND_1	24:56,157	9,79%	SPRT_80_ND_5	09:25,361	12,35%
SPTi_160_ND_5	27:51,973	9,64%	SPTi_80_ND_1	12:19,668	10,19%

(a)

(b)

QUADRO 6.4 – CLASSIFICAÇÃO DOS 10 MELHORES (A) TEMPOS MÉDIOS E (B) ERROS MÉDIOS DO PROBLEMA LA21: ORDENAÇÃO BASEADA NO BALANÇO TEMPO X ERRO

A formação contendo SPTi, PM igual a 5% e população menor, despontou como sendo a melhor relação tempo x erro%, seguida pelas conjunturas semelhantes, mas com PM de 1% ou então contendo a heurística RHC, e PM de 5%, para mesmo tamanho de população.

#### 6.1.2.5. Discussões gerais sobre os ensaios realizados

Neste âmbito, já delineando os comentários globais observados

nos ensaios, embora tenham sido calculados, os agendamentos A e ND apresentaram diferenças muito grandes em termos de função objetivo. Quando estabelecido um ponto de corte para elencar as 10 melhores soluções (o ponto de corte foi considerar apenas as soluções que fossem menores que a metade do valor máximo obtido) para facilitar as análises, as formações que focavam em agendamentos sem atraso (*non-delay*) obtiveram convergências mais rápidas, mas principalmente, erros mínimos e erros médios muito inferiores aqueles das formações com agendamentos ativos apenas, e foram desclassificadas das análises.

Isto pode ser explicado considerando-se que, apesar dos agendamentos ótimos estarem contidos no conjunto dos agendamentos ativos, este conjunto é consideravelmente grande e ainda contém agendamentos com tempos relativamente grandes de processamento, considerando-se a função objetivo como sendo o *makespan* (GONÇALVES ET AL, 2005; ARTIGUES E LOPES, 2000). Como o conjunto dos agendamentos Ativos- ND estão inseridos no conjunto dos agendamentos ativos e primam por menos ociosidade dos equipamentos, torna-se evidente que as soluções destes sejam mais otimizadas. Contudo, como Mattfeld e Bierwirth (2004), esperava-se encontrar soluções ativas que fossem melhores que as Ativas ND, mas obviamente este AG, como o dos autores, apesar dos bons resultados, falha ao explorar o grande espaço de busca dos agendamentos ativos.

Outro ponto que merece destaque é o tamanho da população. Como uma população define um espaço de busca, quanto maior ela for, maior será também este espaço. Considerando que os cálculos são aplicados sobre os indivíduos, quanto maior a população, proporcionalmente maior será o tempo computacional gasto na busca exploratória. Torna-se interessante, portanto, avaliar se as populações menores permitem obter resultados com mesma qualidade que as populações maiores. Observa-se que, para a mesma classe de agendamento e com mesma taxa de mutação, praticamente não há alteração nos valores médios e mínimos encontrados, contudo o tempo para atingir o critério de parada eleva-se quanto maior for sua dimensão (alcançar 2000 gerações com uma população pequena é mais rápido que com uma população grande).

Buscando-se definir um consenso entre heurísticas, taxas de PM e tamanho de população, pode-se adiantar que não é tarefa fácil, haja visto naturezas diferentes que cada tipo de problema pode apresentar. Isto é comprovado em cada ensaio feito. Para um, com determinadas características de dimensão e tempos de processamento, desponha uma conjuntura e definem-se tempos e erros conforme o algoritmo atua.

Dentro de uma visão que almeja oportunizar o balanço entre menor tempo de processamento e menores erros em relação ao ótimo (se conhecido), serão elencadas as conjunturas que obtiveram este tipo de eficiência junto aos problemas propostos.

É consenso que a heurística RND despontou em todos os ensaios feitos, em uma instância ou outra, mas também foi destaque a SPTi e a RHC. Também foram destaques as populações com menores dimensões, embora em alguns casos os melhores resultados tenham sido atingidos com populações maiores e taxas de PM baixas. Assim sendo, não foi focado valores instantâneos, mas sim valores médios para composição da análise tempo x erro, buscando desta forma uma maior constância do método sobre um tipo de problema/situação.

Com base nos esquemas (heurísticas, tamanho de população, taxa PM e forma de agendamento) presentes nos Quadros 6.1 a 6.4, buscou-se organizá-los segundo critério de média de pontos, atribuindo a escala mostrada no Quadro 6.5. Abrangendo todos os testes, objetivou-se selecionar aquelas conjunturas mais expressivas.

Lugar	Pontos
1	15
2	12
3	10
4	8
5	6
6	5
7	4
8	3
9	2
10	1
Não apareceu	0

FONTE: PRÓPRIA

QUADRO 6.5 – DESIGNAÇÃO DOS PONTOS PARA ORDENAÇÃO EM FUNÇÃO DA ORDEM

Na Tabela 6.1 se tem esta ordenação, promovendo-se conclusões quanto aos tempos médios.

Além das observações empíricas já realizadas, constata-se por esta organização que a heurística SPTi (seguida da RHC e RND) tem forte presença junto às melhores posições nos testes, quesito tempo médio. Não obstante, as formações contendo baixas populações também é muito frequente entre os melhores resultados. Já quanto às taxas de PM, nota-se tênue preferência para probabilidades mais elevadas.

Média de pontos					
Esquema	LA02	ABZ6	LA21	FT10	Md PTS
SPTi_50_ND_5	0	15	15	15	11,25
RHC_50_ND_5	12	8	5	8	8,25
SPTi_50_ND_1	8	12	1	10	7,75
RND_50_ND_5	15	2	12	2	7,75
RHC_50_ND_1	6	1	8	6	5,25
SPTi_100_ND_1	5	0	0	12	4,25
SPRT_50_ND_5	4	10	2	0	4,00
SPTq_50_ND_1	1	3	10	0	3,50
SPTq_50_ND_5	0	4	6	0	2,50
RND_50_ND_1	10	0	0	0	2,50
SPRT_100_ND_1	0	5	4	0	2,25
SPRT_100_ND_5	0	6	0	0	1,50
RND_100_ND_1	0	0	0	5	1,25
SPTi_100_ND_5	0	0	0	4	1,00
RHC_100_ND_5	2	0	0	1	0,75
RHC_100_ND_1	0	0	0	3	0,75
SPTq_100_ND_5	0	0	3	0	0,75
SPRT_50_ND_1	3	0	0	0	0,75
RND_100_ND_5	0	0	0	0	0,00
SPTq_100_ND_1	0	0	0	0	0,00

FONTE: PRÓPRIA

TABELA 6.1 –JSSP CLÁSSICOS: ORDENAÇÃO DECRESCENTE EM FUNÇÃO DO TEMPO MÉDIO E MÉDIA DE PONTOS

O mesmo procedimento foi realizado ao organizar os dados segundo o erro médio frente aos resultados ótimos. Na Tabela 6.2 estão ordenados decrescentemente, conforme pontuação (vide Quadro 6.6).

Média de pontos					
	LA02	ABZ6	LA21	FT10	Md PTS
RND_50_ND_5	15	8	15	3	10,25
SPTi_50_ND_5	10	0	5	12	6,75
SPRT_50_ND_5	12	5	10	0	6,75
SPTi_50_ND_1	8	1	2	15	6,50
SPTi_100_ND_1	0	10	12	0	5,50
RHC_50_ND_1	4	2	3	6	3,75
RHC_100_ND_1	0	15	0	0	3,75
RHC_50_ND_5	3	4	6	1	3,50
SPRT_50_ND_1	0	12	0	0	3,00
SPTq_50_ND_1	5	0	1	5	2,75
SPTq_50_ND_5	2	0	8	0	2,50
RND_50_ND_1	0	0	0	10	2,50
RND_100_ND_1	0	0	0	8	2,00
RHC_100_ND_5	0	3	0	4	1,75
SPRT_100_ND_1	0	0	4	2	1,50
SPTi_100_ND_5	6	0	0	0	1,50
SPTq_100_ND_5	0	6	0	0	1,50
SPRT_100_ND_5	1	0	0	0	0,25
RND_100_ND_5	0	0	0	0	0,00
SPTq_100_ND_1	0	0	0	0	0,00

FONTE: PRÓPRIA

TABELA 6.2 – JSSP CLÁSSICOS: ESQUEMAS EM ORDEM DECRESCENTE DE ERRO MÉDIO E MÉDIA DE PONTOS

Tocante ao erro médio, o *ranking* por pontos médios se modifica,

mas em parte. Surge novamente um predomínio da heurística SPTi com populações menores e taxas tanto de 5 como de 1% para PM. Nota-se que as heurísticas RHC não ficam logo a seguir, formando um grupo fechado da 6ª a 8ª posição.

A heurística S/PRT, empatou com a SPTi, para mesmo tamanho de população e taxa PM. Nota-se que ambas compõe a RHC.

Conclui-se que, para problemas clássicos, operar com as regras SPTi, RND e RHC são boas escolhas, quando acompanhadas de populações menores (50% da quantidade de operações, ou próximo disso) e priorizando, suavemente, taxas maiores de mutação, obviamente quando aliadas as demais conjunturas apresentadas durante a realização dos testes (algoritmos genético e GT modificado).

## 6.2 TESTES ENVOLVENDO AUMENTO DE ROTAS ALTERNATIVAS

O problema teste, proposto por Chan (2003), e expresso na Tabela 6.2, serve para avaliar o desempenho dos algoritmos de agendamento e genético na busca por soluções, bem como observar a capacidade conjunta destes ao se expandir a quantidade de rotas.

Para o primeiro teste, são analisados os efeitos envolvendo regras de despacho e políticas de seleção de rotas alternativas. Para este caso, o Chan (2003) obteve tempos próximos de 1000 minutos para suas melhores políticas.

A Tabela 6.4, contudo, aborda o mesmo problema, mas com as modificações realizadas por Araujo (2006), aumentando a flexibilidade de rotas com novas máquinas.

<i>Processo</i>	<i>Operação</i>	<i>Máquina</i>	<i>Pij</i>
1	1	1	105
1	1	2	126
1	2	3	168
1	3	5	70
1	4	2	210
1	4	1	175
2	1	2	140
2	1	3	168
2	2	3	70
2	2	2	112
2	3	5	245
2	4	4	175
3	1	5	200
3	2	1	125
3	3	4	150
3	3	3	135
3	4	2	75
4	1	4	150
4	2	2	150
4	3	5	100
4	4	3	125
4	4	1	75
5	1	1	50
5	2	3	100
5	3	2	75
5	3	5	100
5	4	4	150
6	1	3	175
6	1	5	140
6	2	2	84
6	3	1	175
6	4	5	70
6	4	3	161
7	1	4	245
7	1	1	266
7	2	5	70
7	3	1	70
7	3	4	105
7	4	2	105
8	1	5	105
8	1	4	70
8	2	4	280
8	2	5	210
8	3	3	175
8	4	1	140

FONTE: ADAPTADO CHAN (2003)

TABELA 6.2 – TEMPOS DE PROCESSAMENTO E ROTAS ALTERNATIVAS PROPOSTAS POR CHAN (2003)

<i>Processo</i>	<i>Operação</i>	<i>Máquina</i>	<i>Pij</i>
1	1	1	105
1	1	2	126
1	1	6	91
1	2	3	168
1	2	4	140
1	3	5	70
1	3	3	140
1	4	2	210
1	4	1	175
1	4	4	175
2	1	2	140
2	1	3	168
2	1	7	154
2	2	3	70
2	2	2	112
2	3	5	245
2	3	2	266
2	4	4	175
2	4	5	203
3	1	5	200
3	1	1	100
3	2	1	125
3	2	2	60
3	2	7	60
3	3	4	150
3	3	3	135
3	4	2	75
3	4	4	90
4	1	4	150
4	1	2	75
4	2	2	150
4	2	5	75
4	3	5	100
4	3	4	50
4	3	6	100
4	4	3	125
4	4	1	75
5	1	1	50
5	1	3	40
5	2	3	100
5	2	4	150
5	3	2	75
5	3	5	100
5	3	6	125
5	4	4	150
5	4	3	125

FONTE: ADAPTADO DE ARAUJO (2006)

TABELA 6.3 – TEMPOS DE PROCESSAMENTO E ROTAS ALTERNATIVAS PROPOSTAS POR CHAN (2003) MODIFICADO POR ARAUJO (2006)

## Continuação

<i>Processo</i>	<i>Operação</i>	<i>Máquina</i>	<i>Pij</i>
6	1	3	175
6	1	5	140
6	2	2	84
6	2	4	56
6	2	5	70
6	3	1	175
6	3	4	140
6	4	5	70
6	4	3	161
7	1	4	245
7	1	1	266
7	1	3	245
7	2	5	70
7	2	4	126
7	3	1	70
7	3	4	105
7	3	7	105
7	4	2	105
7	4	5	170
8	1	5	105
8	1	4	70
8	2	4	280
8	2	5	210
8	3	3	175
8	3	2	140
8	4	1	140
8	4	2	56

FONTE: ADAPTADO DE ARAUJO (2006)

TABELA 6.3 (CONT.) – TEMPOS DE PROCESSAMENTO E ROTAS ALTERNATIVAS PROPOSTAS POR CHAN (2003) MODIFICADO POR ARAUJO (2006)

Aplicar os algoritmos sobre estas duas versões do problema de Chan (2003) irá permitir observar o comportamento dos algoritmos frente a um aumento do número de rotas. Como também propôs Araujo (2006), é esperada uma redução no *makespan* com um aumento na flexibilidade de rotas causada pelo aumento no número de máquinas a processar determinadas operações já existentes anteriormente.

O *software* Genesis 2.32 é o ambiente onde o problema, já com a população inicial aleatoriamente construída, é aplicado ao AG e conseqüentemente, ao agendador. Solicitados os ensaios de agendamentos ativo e ativo ND, foram colhidos os resultados de tempo de processamento e função objetivo definida para tempo de processamento.

O *deadline*, ou tempo limite de cada processo, é utilizado pela



heurística RHC e é calculado como sendo o somatório do tempo de cada operação, acrescido de 20%.

Os dados dos testes desta seção estão organizados nos Apêndices F e G, para os ensaios onde o desvio padrão é nulo, ou seja, o programa *Genesis* não considera haver perturbação sobre os tempos de processamento, transporte e *setup* de cada operação.

Tocante às observações realizadas, tem-se que o tempo total de processamento foi reduzido. Como é possível observar, o menor *makespan* médio obtido na solução do problema original foi reduzido de 934 u.t. para 830 u.t. (ou, em valores mínimos encontrados, de 931 u.t. (em 0,884s) para 786 u.t. (em 0,063s)) com tempos médios computacionais bastante próximos e na faixa dos 15 a 20s. Isto era esperado, uma vez que se aumenta o número de rotas e, desta forma, as opções de processamento. Também Araujo (2006) assim constatou em seus testes e levantou a questão de que, uma vez que se aumenta a quantidade de rotas e reduz-se assim o tempo total de processamento, também se reduz a frequência do uso de máquinas, mas há que se observar sobre este investimento, pois a monta do investimento de aquisição de equipamentos poderá ser muito superior ao retorno esperado com a redução destes tempos.

### 6.3 TESTES ENVOLVENDO JSSPS DINÂMICOS

Os testes aplicados na agendamentos dinâmicos não determinísticos NDDJSSP (realistas) necessitaram que fossem introduzidas rotinas de simulação Monte Carlo no algoritmo original, visando, durante o agendamento, gerar variáveis aleatórias provenientes de distribuição normal. A média é o tempo dado e o desvio padrão é definido antes da simulação.

Os tempos simulados são: instante de início de processamento da operação; tempo de transporte entre máquinas; tempo de processamento da operação na respectiva máquina. Todos estes são fundamentais para definir o tempo final de cada operação, de cada processo.

Para os testes, dois foram os problemas escolhidos da literatura: o proposto por Kumar et al (2003) e o proposto por Chan (2003). Tendo posse das modificações provenientes de Araujo (2006) sobre o problema proposto por CHAN, como já discutido na seção 6.2, aplicou-se também sobre este as oscilações (ou turbulências) mencionadas.

Originalmente o problema proposto por Kumar et al (2003) é solucionado por um algoritmo heurístico baseado em colônias de formigas com o intuito de solucionar agendamentos em manufatura

flexível. Araujo (2006) aponta que o tempo total de processamento varia entre 41,7 e 43,2s para demandas unitárias de produção, concluindo que a melhor solução encontrada é fruto da escolha, dentre diversas rotas, de apenas uma única, conforme o *makespan*.

<i>Processo</i>	<i>Operação</i>	<i>Máquina</i>	<i>Pij</i>
1	1	1	7
1	1	3	4
1	2	2	3
1	3	1	3
1	3	3	6
1	4	1	2
1	4	2	4
2	1	1	8
2	1	2	12
2	2	3	4
2	3	1	7
2	3	2	14
2	4	1	8
2	4	3	4
3	1	1	10
3	1	2	15
3	1	3	8
3	2	2	2
3	2	3	6
3	3	1	2
3	3	3	4
3	4	1	6
3	4	2	3
4	1	2	9
4	1	3	5
4	2	1	6
4	2	3	2
4	3	2	7
4	3	3	12
4	4	1	9
4	4	2	6
4	4	3	3
5	1	1	10
5	1	3	15
5	2	2	7
5	2	3	14
5	3	1	5
5	3	2	8
5	4	1	4
5	4	2	6
5	4	3	8

FONTE: ADAPTADO DE ARAUJO (2006) APUD KUMAR ET AL (2003)  
TABELA 6.4 – TEMPOS DE PROCESSAMENTO E ROTAS  
ALTERNATIVAS PROPOSTAS POR Kumar et al (2003)

Dentro das propostas desta tese, os testes simulam o agendamento de processos turbulentos pela oscilação dos tempos ou variável aleatória, oriunda de distribuição normal.

Para gerar estas variáveis aleatórias, a função desenvolvida necessita dos valores da média e desvio padrão, sendo a média estabelecida pelo problema original e o desvio padrão variando experimentalmente em 0, 5, 10 e 20%. Tais parâmetros são inseridos junto aos dados iniciais, no programa Genesis.

O processo de simulação é visto como um gerador de turbulências, criando indeterminismos no decorrer do agendamento. Oscilar os dados de tempo de *setup* (dependente), tempo de processamento, tempo de início e tempo de transporte é importante para avaliar o desempenho na solução do NDDJJSP, haja visto reagir no momento em que é detectada a mudança, adaptando-se instantaneamente à informação dada pelo sistema.

### 6.3.1 Resultados e discussões

#### 6.3.1.1 Problema CHAN original

Conforme definido por Chan (2003) e expressos na Tabela 6.2, o problema foi agendado e simulado considerando as premissas de turbulência: 0, 5, 10 e 20%.

Para as médias, da função objetivo e do tempo computacional gasto, é possível constatar sua grande proximidade, não havendo grandes diferenças entre escolher uma ou outra heurística para agendamento. Também os tempos diferem pouco dentro de cada classe de oscilação analisada. Com tempos médios entre 30s e 1 minuto, é possível encontrar bons resultados, em média, independente da heurística aplicada, PM ou forma de agendamento.

Já mencionado, Chan (2003) encontrou valores próximos de 1000 unidades de tempo (u.t.) para o ensaio realizado sem turbulência. Nos ensaios simulados, os dados aproximados mínimos e médios, para cada caso são: 931 e 934, 960 e 990, 1013 e 1045, 995 e 1022, respectivamente para as simulações com turbulências de 0 (sem variação), 5, 10 e 20%. É visto existe correlação de amplitude e as taxas de perturbações geradas, mas variando muito pouco de uma classe para outra, o que demonstra boa reação durante processo de agendamento no que se refere à ambientes dinâmicos.

Se traçado um breve comparativo entre os valores de *makespan* obtidos pelos autores e nesta tese, é possível constatar que os últimos

são inferiores, mostrando que conjuntura adotada é eficiente na condução das soluções para os problemas e que, ao menos a hipótese de conjugar um número maior de rotas é, além de viável na busca das soluções, também vantajosa sob o aspecto de conduzir a respostas mais rápidas e melhores. Expressando valores, Chan (2003) apresenta um *makespan* próximo de 1000 u.t.; Araujo (2006) atinge valores médios próximos de 1070 u.t.; a conjuntura trabalhada nesta tese atinge o menor valor igual a 931u.t. e um valor médio de 934u.t. (aprox. 6,9% menor).

### 6.3.1.2 Problema CHAN modificado

Com seus dados dispostos na Tabela 6.3, o problema foi submetido à sequência de ensaios já definida. Os resultados foram tabulados em planilhas apresentadas em Apêndices, G.

Como critério de avaliação, definiu-se um índice que relacione tempo computacional médio e valor da função objetivo (*makespan*). O objetivo é ordenar de forma crescente os dados obtidos, segundo este índice. Este *ranking* permite observar quais combinações são mais promissoras nos dois quesitos.

Na figura G.1, em Apêndices, é possível observar a grande proximidade entre os diferentes valores para a função objetivo, seja quando observados para cada classe de perturbação, seja quando observados na totalidade. Também os tempos, se observados os cinco melhores do *ranking*, são gastos em média, não mais que 45s para as classes de oscilação de 5, 10 e 20%, mas 23s para as mesmas condições quando estudado o problema sem qualquer perturbação nos tempos.

Assim, embora em ordenação crescente considerando tempos de processamento e baixos valores da função objetivo, a heurística RHC mostrou-se eficiente, ficando entre as 4 primeiras posições nos testes, liderando 50% deles. Outras que obtiveram destaque foram a SPTq (maior frequência entre as 5 melhores posições), SPTi e a RND.

Como pode ser visto em Apêndices, na Figura G.1 (e G.2) as formações considerando agendamentos ND apresentam, para a função objetivo selecionada, menores resultados que as de formação A, embora haja bastante equilíbrio quando observados os tempos médios.

### 6.3.1.3 Problema Kumar

Como foram desenvolvidos os experimentos sobre os problemas anteriores, também o apresentado por Kumar et al (2003) foi submetido a variações de tempo e os resultados não distaram muito do esperado.

Quanto aos dados obtidos em trabalhos anteriores, o *makespan* expresso pelos autores em um gráfico fica entre 41,7 e 43,2s. Aqui, adotar-se-á 41,7s como menor valor encontrado por Kumar et al (2003).

Araujo (2006), aplicando os mesmos dados, alcançou o valor de 410s para 10 peças, logo 41s médios para cada peça, demonstrando que o resultado da função objetivo de Kumar et al (2003) não era ótimo.

Empregando mesma demanda que Araujo (2006), ou seja, 10 peças, foi possível alcançar um *makespan* mínimo de 350s, ou seja, 35s médios para cada peça. Este resultado foi bastante animador, pois demonstra a boa aplicabilidade dos algoritmos presentes nesta conjuntura para problemas com multiplicidade de rotas.

Tocante a análise de desempenho frente a perturbações de ordem 5, 10 e 20% sobre os tempos de processamento, *setups* e de início de cada operação, observa-se que os algoritmos aplicados demonstraram robustez, haja visto que os tempos finais médios pouco oscilaram em relação ao tempo de 350s. Os gráficos e demais dados referentes às aplicações das perturbações podem ser vistos em Apêndices H. Respectivamente às perturbações mencionadas, têm-se os *makespans* médios: 381, 424, 501s, com tempos computacionais de 14, 17 e 21s.

#### 6.4 TESTES ENVOLVENDO AGENDAMENTO COM TEMPOS DE *SETUP* DEPENDENTES

A fim de poder avaliar o efeito do agendamento considerando os tempos de *setup* dependentes, toma-se o exemplo de Yamada e Nakano (1997), já ilustrado na Figura 3.2 e com os dados contidos na Tabela 3.4. O objetivo é promover um comparativo deste problema clássico com tempos de *setup* nulos e não nulos. Na seção 6.5.1 tem-se um exemplo detalhado de agendamento ativo. Na seção 6.5.2 é apresentado um agendamento ativo sem atraso, também para o mesmo problema.

##### 6.4.1. Agendamento com tempos de *setup* dependentes: demonstrando o funcionamento do algoritmo GT e genético

Selecionado um indivíduo para análise do agendamento ativo promovido pelo algoritmo GT, temos dele as seguintes informações:

JOB	OPER.	MAQ.	ORDEM	$T_{início}$	$T_{final}$	$P_{ij}$
3	3	3	0	0	1	1
3	1	2	1	0	3	3
1	1	1	2	0	3	3
2	3	2	3	0	4	4
2	2	3	4	0	3	3
1	3	3	5	0	3	3
1	2	2	6	0	3	3
2	1	1	7	0	2	2
3	2	1	8	0	2	2

TABELA 6.4 – DADOS DE UM INDIVÍDUO FORMADO ALEATORIAMENTE SEGUNDO INFORMAÇÕES DO PROBLEMA DE AGENDAMENTO DE TAREFAS EM JOB-SHOP 3X3 PROPOSTO POR YAMADA E NAKANO (1997)

A única informação que se altera nos indivíduos dos problemas clássicos é a ordem ou sequência atribuída a cada operação, dado que as máquinas que irão processar cada tarefa já são previamente conhecidas.

O procedimento de agendamento pressupõe formar, basicamente, os conjuntos C, G e A, contendo respectivamente as informações candidatas, as eleitas para o conjunto conflito e as efetivamente agendadas. Das operações candidatas é escolhida a operação O', ou seja, aquela que tem o tempo de finalização mais cedo. O Conjunto conflito G é formado com base nas operações que disputam com O' o mesmo recurso, ou máquina M'. Caso exista esta disputa, a forma de eleger a operação a ser agendada, ou O\*, é por meio da ordem definida na estrutura (“corpo”) do cromossomo.

Escolhidas as operações iniciais de cada tarefa para serem as operações candidatas, parte-se por desenvolver algumas iterações do algoritmo GT modificado.

Exemplo de aplicação do algoritmo:

Iteração 1:

$C = \{(3,1,2-0,3); (1,1,1-0,3); (2,1,1-0,2)\};$

$C = \min. \{3; 3; 2\} \rightarrow O' = (2,1,1)$

$M' = \text{máquina de } O' = 1$

$G = \{(1,1,1); (2,1,1)\};$

$O^* = \{(1,1,1)\};$

$A = \{(1,1,1 - 0,3)\}$

$G = \{(2,1,1 - 3,5)\}$

$C = \{(3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6)\};$

Iteração 2:

$$C = \{(3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6)\};$$

$$C = \min. \{3; 5; 6\} \rightarrow O' = (3,1,2)$$

$M' = \text{máquina de } O' = 2$

$$G = \{(3,1,2 - 0,3); (1,2,2 - 3,6)\};$$

$$O^* = \{(3,1,2 - 0,3)\};$$

$$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3)\}$$

$$G = \{(1,2,2 - 3,6)\};$$

$$C = \{(2,1,1 - 3,5); (1,2,2 - 3,6); (3,2,1 - 3,5)\};$$

Iteração 3:

$$C = \{(2,1,1 - 3,5); (1,2,2 - 3,6); (3,2,1 - 3,5)\};$$

$$C = \min. \{5; 6; 5\} \rightarrow O' = (2,1,1) \text{ ou } (3,2,1) \rightarrow \text{menor ordem} \rightarrow O' = (2,1,1 - 3,5)$$

$M' = \text{máquina de } O' = 1$

$$G = \{(2,1,1 - 3,5); (3,2,1 - 3,5)\};$$

$$O^* = \{(2,1,1 - 3,5)\};$$

$$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3); (2,1,1 - 3,5); \}$$

$$G = \{(3,2,1 - 5,7)\};$$

$$C = \{(1,2,2 - 3,6); (3,2,1 - 5,7); (2,2,3 - 5,8)\};$$

Iteração 4:

$$C = \{(1,2,2 - 3,6); (3,2,1 - 5,7); (2,2,3 - 5,8)\};$$

$$C = \min. \{6; 7; 8\} \rightarrow O' = (1,2,2)$$

$M' = \text{máquina de } O' = 2$

$$G = \{(1,2,2 - 3,6)\};$$

$$O^* = G = \{(1,2,2 - 3,6)\};$$

$$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6)\}$$

$$G = \{ \};$$

$$C = \{(3,2,1 - 5,7); (2,2,3 - 5,8); (1,3,3 - 6,9)\};$$

Iteração 5:

$$C = \{(3,2,1 - 5,7); (2,2,3 - 5,8); (1,3,3 - 6,9)\};$$

$$C = \min. \{7; 8; 9\} \rightarrow O' = (3,2,1)$$

$M' = \text{máquina de } O' = 1$

$$G = \{(3,2,1 - 5,7)\};$$

$$O^* = G = \{(3,2,1 - 5,7)\};$$

$$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6); \{ (3,2,1 - 5,7) \};$$

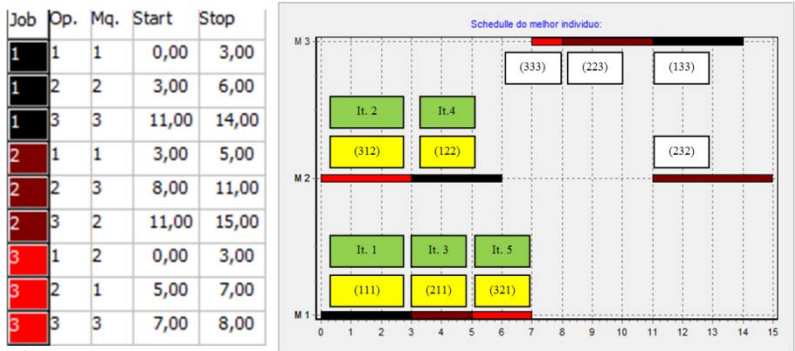
$$G = \{ \};$$

$C = \{ (2,2,3 - 5,8); (1,3,3 - 6,9); (3,3,3 - 7,8) \};$

Para estas 5 primeiras iterações, o agendamento fica:

$A = \{ (1,1,1 - 0,3); (3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6); \{ (3,2,1 - 5,7) \};$

O gráfico de Gantt ilustra o agendamento total, aqui realizado até a iteração 5:



FONTE: PRÓPRIA

FIGURA 6.2 – RECORTES DE TELA DO PROGRAMA GENESIS 2.32: AGENDAMENTO ATIVO PARA PROBLEMA YN97 SEM TEMPOS DE *SETUP* DEPENDENTES

Considerando o mesmo indivíduo, inclusive com mesma designação de ordem para as operações, mas com tempos de setup dependentes (recurso e antecessora), definidos nas matrizes:

	J(1,1)	J(2,1)	J(3,2)
J(1,1)	0	3	0
J(2,1)	0	0	0
J(3,2)	0	0	0

FONTE: PRÓPRIA

FIGURA 6.3 – RECORTES DE TELA DO PROGRAMA GENESIS 2.32: ENTRADA DE DADOS DE TEMPOS DE *SETUP* DEPENDENTES

Observa-se que se 211 seguir a operação 111 na máquina 1, terá  $t_{setup}=3$  u.t. para as demais operações, nas demais máquinas, os tempos de setup são nulos.



Tem-se o agendamento para as cinco primeiras iterações:

Iteração 1:

$C = \{(3,1,2-0,3); (1,1,1-0,3); (2,1,1-0,2)\};$

$O' = \min. \{3; 3; 2\} \rightarrow O' = (2,1,1)$

$M' = \text{máquina de } O' = 1$

$G = \{(1,1,1); (2,1,1)\};$

$O^* = \{(1,1,1)\};$

$A = \{(1,1,1 - 0,3)\}$

$G = \{(2,1,1 - 3,5)\}$

$C = \{(3,1,2 - 0,3); (2,1,1 - 3,5); (1,2,2 - 3,6)\};$

Iteração 2:

$C = \{(3,1,2 - 0,3); (2,1,1 - 3,8); (1,2,2 - 3,6)\};$

$O' = \min. \{3; 8; 6\} \rightarrow O' = (3,1,2)$

$M' = \text{máquina de } O' = 2$

$G = \{(3,1,2 - 0,3); (1,2,2 - 3,6)\};$

$O^* = \{(3,1,2 - 0,3)\};$

$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3)\}$

$G = \{(1,2,2 - 3,6)\};$

$C = \{(2,1,1 - 3,5); (1,2,2 - 3,6); (3,2,1 - 3,5)\};$

Iteração 3:

$C = \{(2,1,1 - 3,8); (1,2,2 - 3,6); (3,2,1 - 3,5)\};$

$O' = \min. \{8; 6; 5\} \rightarrow O' = (3,2,1)$

$M' = \text{máquina de } O' = 1$

$G = \{(2,1,1 - 3,5); (3,2,1 - 3,5)\};$

$O^* = \{(2,1,1 - 3,5)\};$

$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3); (3,2,1 - 3,5); \}$

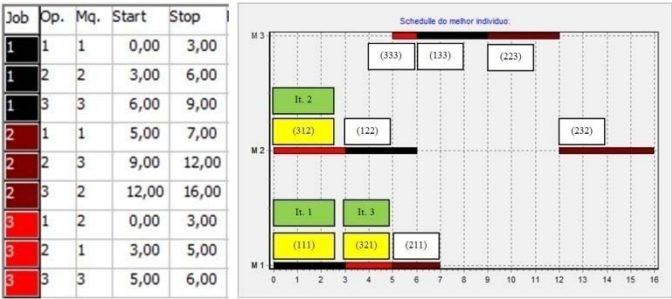
$G = \{(2,1,1 - 5,8)\};$

$C = \{(2,1,1 - 5,8); (1,2,2 - 3,6); (3,2,1 - 5,7)\};$

Para estas 3 primeiras iterações, o agendamento fica:

$A = \{(1,1,1 - 0,3); (3,1,2 - 0,3); (3,2,1 - 3,5); \}$

O gráfico de Gantt ilustra o agendamento total, aqui realizado até a iteração 3:



FONTE: PRÓPRIA  
FIGURA 6.4 – RECORTES DE TELA DO PROGRAMA GENESIS 2.32: AGENDAMENTO ATIVO PARA PROBLEMA YN97 COM EFEITO DOS TEMPOS DE *SETUP* DEPENDENTES

O tempo de *setup* dependente da operação antecessora fez com que o agendador optasse por não escolher a operação (2,1,1), elegendo a candidata (3,2,1). Por ser dependente de uma antecessora, o tempo de processamento aumenta de 2 u.t. para 5 u.t. Assim, o tempo de finalização passaria de 5u.t. para 8u.t. Como o processo de decisão do agendamento ativo é sensível ao menor tempo de finalização, a tarefa (2,1,1) fica em segundo plano na designação da operação O', e conseqüentemente na formação do conjunto conflito G.

A seguir tem-se o agendamento Ativo-ND com tempos diferenciados para cada transição entre operações, para cada máquina.

6.4.2. Agendamento em formação ND considerando diversos tempos de *setup* dependentes

Apesar de realizada com menor detalhamento dado o grande número de iterações, nesta seção é apresentado o teste envolvendo o problema da seção anterior (YN97 de Yamada e Nakano (1997), embora aqui com tempos de *setup* dependentes, iguais para cada uma das máquinas, expressos no Quadro 6.5.

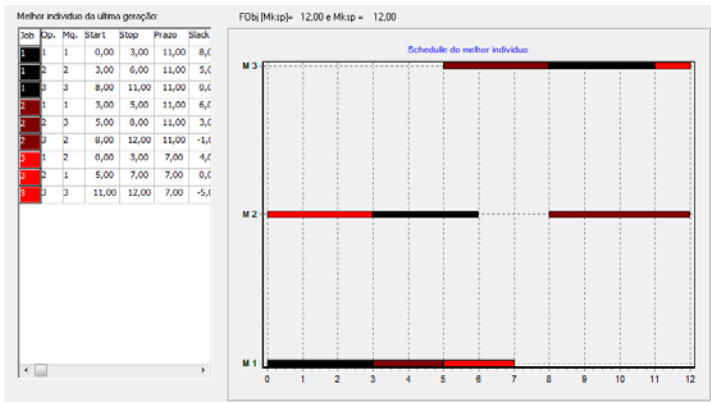
		Jobs (destino)		
		1	2	3
Jobs (origem)	1	0	2	3
	2	4	0	5
	3	6	7	0

FONTE: PRÓPRIA  
QUADRO 6.6 – TEMPOS DE *SETUP* DEPENDENTES PROPOSTOS PARA AS TRÊS MÁQUINAS DO YN97

Como não existem transições para mesmos nós e o problema tem cunho meramente ilustrativo, assumiu-se valores inteiros e sequenciais de 2 a 7, dispostos em sequencia crescente nas células.

Os tempos de *setup* dependentes levam o agendador à tomada de decisão aliando estes aos tempos de transporte e tempos de processamento de operações, para cada máquina.

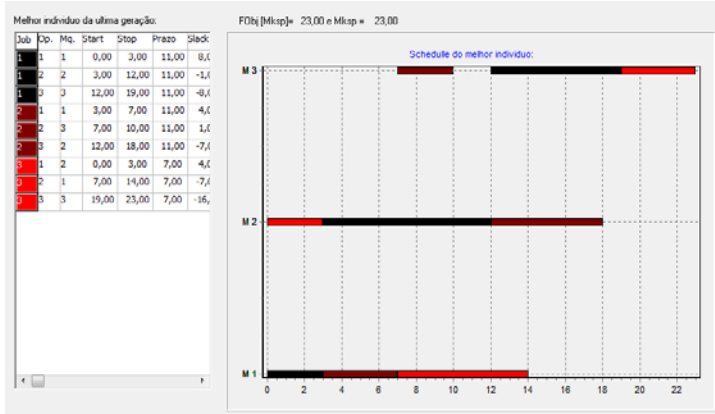
Para comparação, tem-se o gráfico de Gantt, na Figura 6.5.



FONTE: PRÓPRIA

FIGURA 6.5 –TELA DO PROGRAMA GENESIS: AGENDAMENTO ATIVO ND PARA PROBLEMA YN97 SEM TEMPOS DE *SETUP* DEPENDENTES

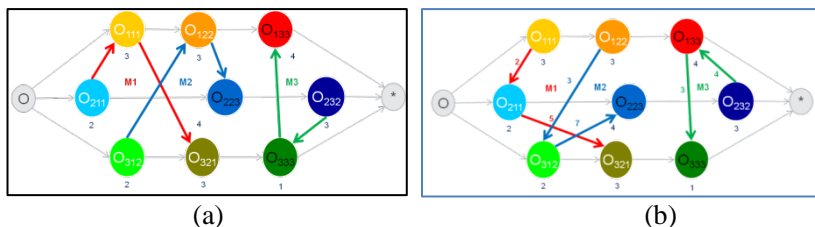
Na Figura 6.6, tem-se o YN97 modificado, contendo tempos de *setup* dependentes.



FONTE: PRÓPRIA

FIGURA 6.6 –TELA DO PROGRAMA GENESIS: AGENDAMENTO ATIVO ND PARA PROBLEMA YN97 COM TEMPOS DE *SETUP* DEPENDENTES

Os respectivos grafos das soluções apresentadas são representados nas Figuras 6.7 e 6.8, onde é possível observar o efeito dos tempos de *setup* dependentes sobre a tomada de decisão do agendador, também para problemas de formação ativa ND.



FONTE: BASEADO NO GRAFO DE YAMADA E NAKANO (1997)

FIGURA 6.7 – GRAFO REPRESENTANDO AGENDAMENTO ATIVO ND PARA PROBLEMA YN97 (A) SEM E (B) COM TEMPOS DE *SETUP* DEPENDENTES

Como os resultados foram obtidos com o auxílio do *software* Genesis, uma vez que aplica os algoritmos propostos, constata-se sua capacidade de solucionar os problemas também das situações NDDJSSP

## 6.5 TESTES ENVOLVENDO AGENDAMENTO EM PROBLEMAS COM QUEBRAS DE MÁQUINAS

Para avaliar o comportamento quanto a quebras de máquinas, o programa Genesis permite emular a parada/quebra de determinada máquina durante a formação de um agendamento. Em suma, significa reservar, em tempo de execução, o referido recurso com a maior prioridade possível. O equipamento fica “em manutenção” durante período especificado (o tempo pode ser uma variável aleatória proveniente de distribuição normal com média e desvio padrão definidos pelo usuário) e posteriormente é liberado.

O intuito é promover uma quebra por longo período de um dado recurso, fugindo totalmente de uma leve perturbação abordada nos testes de NDDJSSP (oscilações), onde poderá ocorrer quebra de produção e o processo deverá ter continuação após manutenção ou prosseguir em outro recurso, se for possível.

Desta forma, é objetivo avaliar se o algoritmo é capaz de adaptar-se no instante da quebra e continuar a operação em outra rota. Isso é esperado, uma vez que a base para este re-agendamento é o próprio algoritmo GT modificado aplicado nos demais ensaios, contudo com

modificações para re-agendamento a partir do instante da quebra.

Observa-se que esta característica de re-agendamento obriga o sistema a operar com apenas uma solução, explorando as possibilidades apenas daquelas máquinas ainda não agendadas até a ocorrência do evento de quebra.

Pode-se imaginar que um simples atraso nas operações é satisfatório, mas se a quebra ocorre sobre uma máquina gargalo (maior frequência de uso, por exemplo) o atraso poderá ser muito maior que o tempo de parada para manutenção e tempos de setup originários da nova alocação da operação com quebra de produção, surgindo neste ponto à vantagem de se re-agendar a situação pós-quebra.

Tocante aos procedimentos aplicados nos testes, inicialmente são conduzidos os passos já aplicados aos demais problemas, parametrizando os algoritmos através do Genesis, até que uma solução satisfatória seja encontrada.

Para simular uma quebra de equipamento/produção, vale-se desta solução. O algoritmo a seguir descreve com mais detalhes, contudo discorre-se brevemente sobre os procedimentos a adotar.

Como é feita um inserção de novas atividades no problema já definido, é preciso modificar o indivíduo que denota a melhor solução, configurando novas operações: operação até a quebra e operação manutenção. A operação que, eventualmente for interrompida pela quebra de recurso terá dois momentos: antes da quebra e depois da quebra. Antes da quebra, torna-se outra operação, denominada operação extra, ou operação até a quebra. Já o que necessita ainda ser processado ficará vinculado ao nome original da operação, codificado internamente no programa. Tocante a operação de manutenção, esta contém os dados de início e término da parada, bem como a máquina em que isso ocorre.

Depois de modificado, o indivíduo é submetido à análise, onde são selecionadas todas as operações que necessitam ser fixadas no agendamento, haja visto que já estavam em andamento quando ocorreu a quebra. Salienta-se que a operação em que ocorre o evento, embora também identificada, é fracionada em duas, como já mencionado e, o restante, postergado para ser agendado com as demais, ainda não processadas.

A partir deste ponto, os procedimentos são iguais ao agendamento usualmente aplicado nos demais problemas.

A seguir, tem-se o algoritmo GT modificado, com alteração para processamento de quebras de máquina.

***Algoritmo G&T\_modificado para casos de quebra (A ou A-ND)Início***

**Passo 1:** Coletar dados de quebra: máquina ( $M_{\text{quebra}}$ ), tempo de início ( $t_{iq}$ ) e duração da manutenção ( $p_{ijq}$ );

**Passo 2:** De posse das operações configuradas no indivíduo, fixar (agendar) aquelas já em processamento no momento da falha: ;

**Passo 3:** Identificar a operação que sofreu quebra de produção, se houver, e removê-la de A:

**Passo 4:** Inserir todas as operações sucessoras de cada tarefa contida em A (caso existam) e demais operações iniciais de cada tarefa (do horizonte de planejamento ativo) no conjunto de operações candidatas C;

**Passo 5:** Inserir em A as operações extra e de manutenção: onde oíke tem início em e término em  $t_{ikq}$  e  $O_{ikm}$  tem início em  $t_{ikq}$  e término em  $t_{ikq} + p_{ijq}$ ;

Segue os passos 2 a 9 do algoritmo GT modificado apresentado na seção 5.3.3.9 ou, com as modificações propostas nos passos 2 e 4, se desejado agendamento A-ND, na mesma seção.

**Fim do algoritmo**

Para testes, foram aplicados dois problemas já conhecidos: o proposto por Kumar et al (2003) e o proposto por Chan (2003).

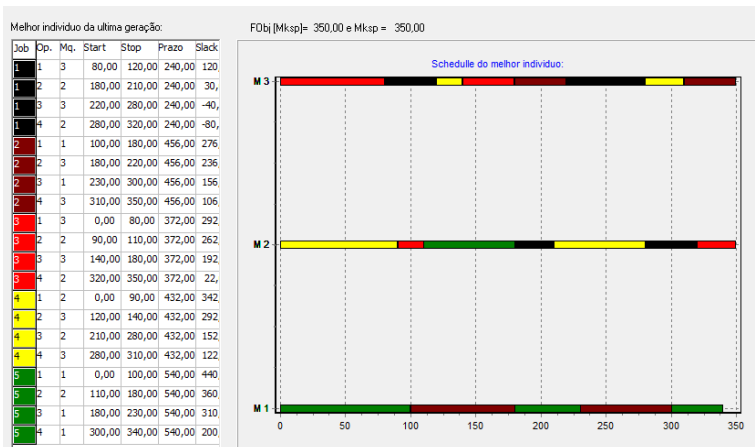
Quanto aos procedimentos de teste, adotou-se a seguinte metodologia: escolher uma boa solução (melhor *makespan* possível) e simular quebra sobre a máquina com maior frequência de uso, a um terço do tempo previamente calculado para processamento total (*makespan*) com duração de manutenção de 60 u.t. (equivalente a 1 hora). A seguir têm-se os testes e os resultados encontrados.

#### 6.5.1 – Teste de quebra de máquina: problema KUMAR

O problema de Kumar et al (2003) foi novamente aplicado, agora para avaliar a capacidade dos algoritmos genético e de agendamento em promover o *schedule* de tarefas mesmo em situações de quebra de máquinas e, provavelmente, também de produção.

No problema original, considerada a heurística RHC, encontrou-

se solução 350 u.t. para a função de desempenho *makespan*, como também o foi para regras SPTi e RND. A seguir tem-se a tela do Genesis indicando o agendamento.



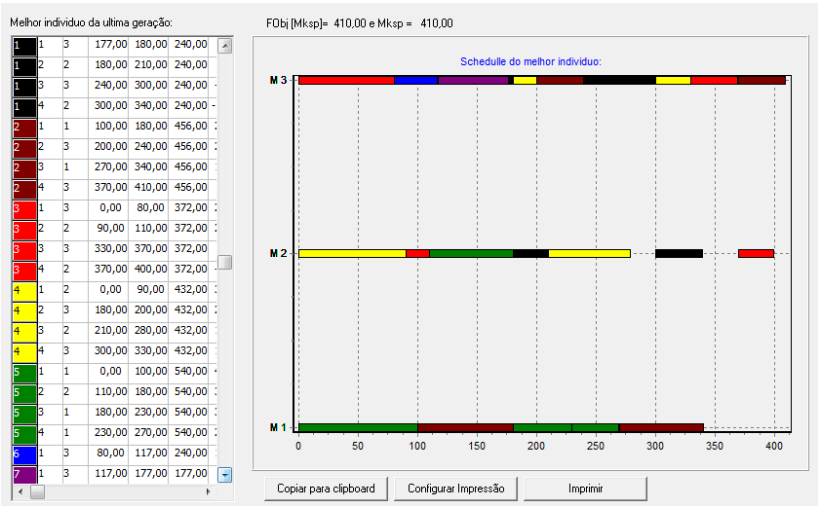
FONTE: PRÓPRIA

FIGURA 6.9 – TELA DO PROGRAMA GENESIS: AGENDAMENTO ATIVO ND DO PROBLEMA KUMAR (KUMAR ET AL, 2003)

Calculada a quebra para ocorrer na máquina com maior frequência de uso (Máquina 3) e decorridos 1/3 do tempo total de processamento (a 117 u.t.), com duração de 60u.t., obteve-se a melhor configuração fornecida pelo algoritmo de agendamento A-ND e heurística também RHC o tempo 410u.t., conforme é possível observar na Figura 6.10.

Este tempo de agendamento final é, coincidentemente, igual à soma do tempo anteriormente obtido mais o tempo de manutenção. De fato, os algoritmos apontam que a uma boa alternativa (não garantindo ser a melhor) dentre outras possíveis para o caso foi apenas postergar as operações dependentes daquelas envolvidas na máquina que apresentou falha.

O *software* também apresentou outras soluções, contudo os tempos foram superiores a este de 410 u.t., e, portanto desconsiderados.



FONTE: PRÓPRIA

FIGURA 6.10 – TELA DO PROGRAMA GENESIS: AGENDAMENTO ATIVO ND DO PROBLEMA KUMAR (KUMAR ET AL, 2003) COM QUEBRA DE MÁQUINA GARGALO.

Outra observação importante é que o autor apresenta o tempo de 410 u.t. como melhor solução frente o método aplicado naquele trabalho. Aqui, obteve-se o mesmo *makespan*, contudo para um agendamento postergado por uma quebra na máquina gargalo com duração de 60 u.t. sobre o melhor agendamento feito (350 u.t.).

O resultado, para uma situação sem quebras de recursos, foi 14,63% menor que o melhor resultado de Kumar et al (2003), provavelmente por considerar todas as possíveis rotas do problema.

Comprova-se neste problema a capacidade integrada dos algoritmos em apresentar boas soluções em tempo computacional aceitável e ainda permitir avaliar o comportamento destes agendamentos frente a quebras que possam vir a ocorrer.

A seguir se apresenta o mesmo teste, agora para o problema de Chan (2003).

6.5.2. Teste de quebra de máquina: problema CHAN original

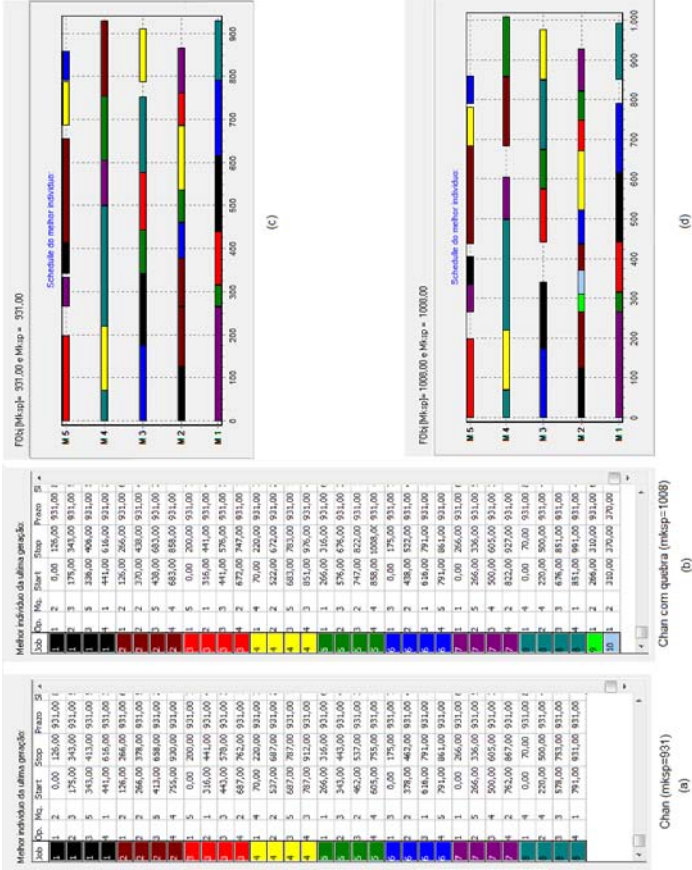
Da mesma forma como realizado par ao problema abordado na



seção anterior, aplicados os algoritmos AG e GT modificado sobre o problema original apresentado por CHAN(2003) tem-se como melhor solução para a função de desempenho *makespan* o valor 931u.t. Assim, introduzida a quebra de 60u.t. em 1/3 do *makespan* encontrado tem-se um novo *makespan*, agora de 1008 u.t.

Na Figura 6.11 estão os gráficos de *Gantt* e as alocações, antes e depois da quebra.

Como na seção anterior, inseridas as operações extra e de manutenção, o algoritmo segue seu curso, visando encontrar soluções que, mesmo com a quebra, venham a minimizar o tempo total de processamento. Observa-se que o simples postergar de tarefas não é elementar ao algoritmo, que tem em sua rotina, inserir operações candidatas conforme regras de despacho previamente definidas. Muitas vezes a escolha de uma ou outra regra pode permitir encontrar soluções melhores ou piores, bem como convergências mais rápidas ou mais lentas. Em geral os resultados obtidos aos testes de submissão de quebra foram satisfatórios e observou-se que o algoritmo apresenta robustez, atendendo aos objetivos propostos no trabalho de permitir gerar agendamentos em ambientes produtivos realistas, considerando quebras de equipamentos/parada de máquinas em problemas NDDJSSP.



FONTE: PRÓPRIA

FIGURA 6.11 – TELAS DO PROGRAMA GENESIS COM AGENDAMENTOS ATIVO ND DO PROBLEMA CHAN (2003): ALOCAÇÃO DE TAREFAS SEM (A) E COM (B) QUEBRA NA MÁQUINA GARGALO E RESPECTIVOS GRAFICOS DE GANTT (C E D).

## 7. CONCLUSÕES FINAIS

O presente trabalho visa propor uma técnica de agendamento empregando regra heurística combinada (RHC) em agendamentos ativos e ativos sem atraso baseados em algoritmo GT (GIFFLER E THOMPSON, 1960) modificado e algoritmos genéticos. O foco é promover agendamentos eficientes em problemas realistas de *job-shop*, que apresentam indeterminismos nos tempos das operações, além de considerar também os tempos de setup dependentes entre as operações em uma mesma máquina.

Combinados e implementados pelo software Genesis V2.32 desenvolvido em linguagem object-pascal (Delphi 2009) foi possível observar o comportamento geral da proposta frente às ambientes turbulentos, demonstrando boa capacidade de agendamento sem contudo ultrapassar os tempos de finalização propostos. Também os bons resultados encontrados frente aos obtidos pelos outros autores tornou ainda mais animadora a aplicação desta conjuntura no auxílio à tomada de decisão.

A regra heurística RHC mostrou-se eficiente junto aos testes realizados, possibilitando obter boas soluções com tempos bastante satisfatórios, provavelmente por ter componentes baseadas em regras igualmente destacadas nos testes: SPT e S/PRT. Independente disso, também a conjuntura adotada para o algoritmo genético colaborou para que indivíduos formados por outras heurísticas pudessem “dopar” em baixas taxas a população que estava sendo construída a cada iteração. Se bons, prevaleceriam, mas se ruins, ainda poderiam ser escolhidos e colaborar com algum segmento útil para novas gerações.

A não existência de super-indivíduos foi importante para que o AG tivesse bom desempenho e selecionasse mais “democraticamente” os indivíduos a cada geração, fruto da seleção proposta por Mayerle (1994).

Em geral, o desempenho foi satisfatório para as aplicações em problemas da literatura, comprovando a capacidade da proposta em organizar, para um dado horizonte de planejamento, as tarefas a serem agendadas em um sistema flexível de manufatura, sejam pertencentes a uma célula de manufatura física ou virtual.

### 7.1 CONTRIBUIÇÕES

Dados os objetivos propostos e os resultados encontrados, este

trabalho promove as seguintes contribuições aos agendamentos em ambientes de manufatura *job-shop* realistas:

Definição de uma técnica de construção de rotas alternativas combinada com o agendamento de tarefas, que considerando as diversas possibilidades de construção de rotas alternativas definidas pelo sistema de manufatura, explore melhor os aspectos reativos de soluções ativas ou ativas sem atraso, escolhendo aquela combinação de rotas que melhor solucione o problema em um dado instante, para certo horizonte de planejamento definido;

Definição de uma estrutura de algoritmo genético que possibilite maior robustez nas soluções diversas de agendamento de *job-shops* realistas, seja durante a etapa de formação da população, com inserções de indivíduos formados por regra gulosa e aleatória, seja na codificação das soluções, com uma estrutura diferenciada para rotas e seqüenciamento;

Definição de um algoritmo modificado de agendamento, facilmente absorvido pelo AG, que permita, além dos agendamentos ativo e ativo sem atraso, considerar tempos dependentes de *setup* e tempos de transporte entre equipamentos e, ainda que seja apto a mudanças repentinas de planejamento, como quebras de máquinas e mudanças de última hora nos tempos que permeiam as operações ainda não agendadas. Aquelas agendadas que sofrem quebras de máquina, o algoritmo sugere novas alocações de tarefas aos recursos, re-agendando-as;

Definição de uma regra heurística combinada que apresente bons resultados e considere prazos de entrega, a fim de integrar-se facilmente nos sistemas de produção sob encomenda e lotes mínimos;

Definição de um *software* que permita implementar os agendamentos de NDDJSSP com tempos de *setup* dependentes e regras heurísticas diversas, conforme a definição de tarefas a serem processadas e disponibilidade, no horizonte de planejamento fixado, dos recursos da célula de manufatura (virtual ou fixa) para processá-las. Este programa permite ainda simular eventos com base em oscilações decorrentes de variável aleatória proveniente de distribuição normal, com parâmetros o AG, agendamento e simulação integrados na interface e definidos pelo usuário.

## 7.2 TRABALHOS FUTUROS

A proposta base deste trabalho versa sobre a aplicação de um algoritmo genético como elemento para promover um agendamento

eficiente e em tempo satisfatório para os *job-shops* com características realistas, ou seja, dinâmicas e não-determinísticas. Para tanto houve o interesse em compor uma regra combinada que aliasse bons desempenhos em casos conhecidos da literatura e a realidade atual, onde a agilidade, flexibilidade, velocidade nas decisões e a qualidade entre cliente-fornecedor é necessária para um bom desempenho da cadeia produtiva: que combinasse a vantagem da regra SPT, que se sobressai na literatura por garantir, em média, bons agendamentos quando desejado menores tempos de processamento total, e outra regra, a S/PRT, onde são considerados o andamento do processamento da tarefa, o prazo de entrega e o quanto de tempo está pendente para finalizar o *job*.

Basicamente a regra foi uma combinação de duas outras com características desejáveis, contudo estudos de composição mais específicos podem ser realizados para formar uma RHC mais reativa a formação de gargalos, sem, contudo incentivar muito escolhas de maior diversidade de máquinas e, assim baixar as frequências de uso dos recursos.

Frente à limitação do trabalho em operar sobre células de manufatura já organizadas e balanceadas, torna-se interessante agrupar esta fase inicial à conjuntura, de forma que a criação da célula virtual (ou fixa) esteja vinculada a certa diversidade de recursos polivalentes disponíveis em dado horizonte de planejamento. Alguns estudos foram propostos por Araujo (2006) e Branco et al (2009). Este último aplicou AGs na formação das células de manufatura, segundo índices de eficácia e eficiência dos agrupamentos.

Dada outra limitação do trabalho, sugere-se considerar a possibilidade de solução com uma visão holística do horizonte de planejamento ao atuar sobre diversas células virtuais balanceadas (e até mesmo criá-las) e apresentar um bom agendamento para todas. Uma boa alternativa pode estar na exploração do paralelismo da conjuntura proposta, que empregaria os mesmos algoritmos, consumindo tempo similar, mas solucionando diversos agendamentos simultaneamente. Sugere-se ainda de forma complementar, a interconexão dos *threads* para compartilharem dos recursos, virtualmente alocados para uma CM e que não foram usados.

Tocante aos agendamentos, o algoritmo GT tem se mostrado eficiente nos seqüenciamentos promovidos, contudo técnicas mais novas tem se evidenciado e merecem atenção. Promover testes comparativos para avaliar se a conjuntura proposta ainda é mais eficiente. Algoritmos construtores como o SSGS – *Serial Schedule Generation Scheme*

proposto por Artigues et al (2005), tem despontado alguns trabalhos, como por exemplo o de Gonzáles et al (2008), que compara esse e uma versão do GT expandido.

Quanto ao algoritmo genético, opções de melhora no esquema proposto podem estar relacionadas aos hibridismos. O projeto atual não aplicou tais inserções, uma vez que o custo computacional de um hibridismo com busca local poderia ser onerado sensivelmente. Entretanto, hibridismos desta forma são comuns em estruturas voltadas para soluções em casos clássicos e poderiam ser melhor estudadas para ampliar a capacidade de busca do método proposto, onde os ganhos podem ser, ironicamente, no tempo, se imprimir resultados mais rápidos. Salienta-se que é uma questão a ser pensada com detalhe, pois como já mencionado, em sistemas realistas, não é esperado um resultado ótimo, mas um bom resultado em tempo razoável.

No nível de *software*, o Genesis está ainda longe de ser uma ferramenta computacional voltada para fins comerciais, mas melhoras sempre serão pretendidas. Algumas podem ser apontadas já na *interface*, desnecessárias para os testes feitos, mas contribuiria para uma melhor análise do usuário, a possibilidade de intervenção direta e de forma gráfica sobre o agendamento em estudo. Poder compor um agendamento envolvendo usuário e algoritmo pode ser interessante, por exemplo, nos casos de quebra de máquina ou atrasos conhecidos na produção. Apesar de possível de ser implementada, esta capacidade não foi incorporada por não ter sido necessário aos testes propostos.

O presente trabalho tem cerne envolto por assuntos perenes, como os agendamentos de tarefas e assim, de problemas combinatórios NP - difíceis, como tendências atuais de flexibilidade na manufatura, produção em lotes mínimos e sob demanda, qualidade durante todo o processo (respeito a prazos), entre outros. Neste âmbito se contribui com nova técnica sob tema bastante discutido e a tendência atual, fruto das necessidades de mercado e produção, cada vez mais voltados ao cliente (ou parceiro). Empregando meios conhecidos é proposta uma nova conjuntura envolvendo algoritmo genético, algoritmo de agendamento e regra heurística combinada para o sequenciamento de tarefas em ambientes turbulentos onde a robustez é necessária. Com atenção aos tempos de recepção e entrega, tolerância a atrasos, quebras e folgas, esta nova proposta contribui no auxílio à tomada de decisão, tão importante na programação e controle da produção de sistemas imersos em uma cadeia produtiva.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, A. M. N., **Escalonamento Dinâmico de Tarefas Industriais sujeitas a prazos de entrega**. Tese de doutorado, Faculdade de Engenharia da Universidade do Porto, Departamento de Eletrotécnica e de computação, Portugal, 1995

APPLEGATE, D., COOK, W., **A computational study of the job-shop scheduling problem**. ORSA Journal on Computing 3, 149–156. 1991.

AYDIN, M. E., FOGARTY, T. C., **A Simulated Annealing Algorithm For Multi-Agent Systems: A Job-Shop Scheduling Application**, Journal of Intelligent Manufacturing, V. 15, N. 6, p. 805-814, 2004.

ARAÚJO JR., L. O., **Método de Programação de Sistemas de Manufatura do Tipo Job Shop Dinâmico Não Determinístico**. Tese (doutorado), Escola Politécnica, Universidade de São Paulo. São Paulo, 2006.

ARTIGUES, C., LOPEZ, P., AYACHE, P., **Schedule generation schemes for the job-shop problem with sequence-dependent setup times: dominance properties and computational analysis**. *Annals of Operations Research* 138:21-52

BARBOZA, A. O., **Simulação e técnicas da computação evolucionária aplicadas à problemas de programação linear inteira mista**, Tese (doutorado), Universidade Tecnológica Federal do Paraná. Paraná, 2005.

BARNES, J.W. e CHAMBERS, J. B., **Solving the Job-Shop Problem with Tabu Search**. IIE Transactions, 27, p. 257-263, 1995.

BECK, F. L., **Escalonamento de Tarefas Job-shop realistas utilizando algoritmos genéticos em Matlab**, Dissertação (mestrado), Universidade Federal de Santa Catarina. Santa Catarina, 2000.

BICKLE,

BIERWIRTH, C., MATTFELD, D., KOPFER, H., **On permutation representations for scheduling problems**. Proceedings of the Fifth International Conference on genetic Algorithms, p. 452-459, Morgan Kaufmann, San Marco, California, 1996.

BIERWIRTH, C., MATTFELD, D., **Production Scheduling and Rescheduling with Genetic Algorithms**, Evolutionary Computation, Volume 7, Número 1, p. 1-16, MIT Press, 1999.

BLANTON, J. L., WAINWRIGHT, R. L., **Multiple vehicle routing with time and capacity constraints using genetic algorithms.** Proceedings of the fifth international conference on genetic algorithms, p. 452-459, Morgan-Kaufmann, san matco, California, 1993.

BRANCO, R. M., COELHO, A. S., MAYERLE, S. F., **Programação da produção: solução de escalonamentos em Job-Shop por algoritmo genético híbrido.** XXVII ENEGEP, 2007, Foz do Iguaçu - PR. Anais do XXVII ENEGEP, 2007

BRANCO, R. M. ; COELHO, A. S. ; MAYERLE, S. F. . **Escalonamentos em Job-shop: solução por algoritmo genético híbrido.** In: 3rd Southern Conference on Computational Modeling, 2009, Rio Grande, RS. Anais do 3rd Southern Conference on Computational Modeling, 2009.

BRAUNE, R., WAGNER, S., AFFENZELLER, M., **Applying Genetic Algorithms to the Optimization of Production Planning in a Real-World Manufacturing Environment.** Cybernetics and Systems 2004, p. 41-46. Austrian Society for Cybernetic Studies, 2004

BRUCKER, P., JURISCH, B., SIEVERS, B., **A branch and bound algorithm for the jobshop scheduling problem.** Discrete Applied Mathematics 49, 109-127, 1994.

BRUKER, P., **Scheduling algorithms.** 2.ed. Springer-Verlag, Berlin, Heidelberg, 1998.

CALIER, J., PINSON, E., **An algorithm for solving the Job-Shop Problem,** Management Science, 35, p. 164-176, 1989.

CANDIDO, M. A. B., **A hybrid genetic approach to solve real make-to-order job-shop scheduling problems.** Tese (doutorado), Universidade Federal de Santa Catarina. Santa Catarina, 1997.

CASTRO, J. P., **Um algoritmo evolucionário para geração de planos de rotas.** Florianópolis, 1999. Dissertação (Mestrado em Engenharia da Produção), Universidade Federal de Santa Catarina.

CHENG, T. C. E., PODOLSKY, S., **Just-in-time manufacturing: an introduction.** 2ª ed. London: Chapman & Hall, 1996.

CHIANG, T-C., FU, L-C., **Using Dispatching Rules for Job Shop Scheduling with Due Date-based Objectives,** Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida - Maio 2006



COELHO, A. S., BRANCO, R. M. e RODRIGUES, G. S., **Aprimoramento da produtividade de sistemas de manufatura e aplicação de algoritmos genéticos na solução de problemas de agrupamentos celulares**, Revista Produção Online, Florianópolis, v.6, n.3, p.116, set./dez., 2006

DIMOPOULOS C., MORT N. A., **Hierarchical Methodology Base don Genetic Programming for the Solution of Simple Cell-formation Problems**. International Journal of Production Research, Vol.39, n.1, p.1-19, 2001

DOMINGOS

FRANCISCHINI, P. G., **Princípios de Gestão da Produção e Logística**. Notas de aula: Engenharia de Produção, USP, 2007. Disponível em: [http://www.pro.poli.usp.br/graduacao/disciplinas-em-andamento/pro2304\\_3/p2304apo\\_sistemas\\_de\\_Producao\\_Versao\\_1.pdf](http://www.pro.poli.usp.br/graduacao/disciplinas-em-andamento/pro2304_3/p2304apo_sistemas_de_Producao_Versao_1.pdf). Acesso em 23 mai. 2007.

FERNANDES, F. C. F., GODINHO FILHO, M., **Sistemas de coordenação de ordens: revisão, classificação, funcionamento e aplicabilidade**. Gestão e produção, São Carlos, V. 14, n. 2, p. 337-352, maio-agosto, 2007.

FRANÇA, P. M., **Notas de aula: EA 043 - Programação da Produção em Sistemas de Manufatura**, Unicamp, Campinas-SP, obtido em 12/08/2007, [www.densis.fee.unicamp.br/~franca/EA043/Transpa-sa.pdf](http://www.densis.fee.unicamp.br/~franca/EA043/Transpa-sa.pdf)

FRENCH, S., **Sequencing and scheduling: an introduction to the mathematics of the job shop**, Ellis Horwood Limited Publishers Chichester, 1982.

GAREY, M., JOHNSON, D., **Computers and intractability: a guide to the theory of NP\_completeness**. W. H. Freeman, Nova Yorque, 1979.

GIFFLER, B., THOMPSON, G., **Algorithms for solving production scheduling problems**. Operations Research, 8: 487-503, 1960.

GLOVER, F., GREENBERG, H. J., **New approaches for heuristic search: a bilateral linkage with artificial intelligence**. European Journal of Operational Research, 39, 119-130. 1989.

GOLDBERG, D. E., **Genetic Algortihms in Search, Optimization, and Machine Learning**, 1 ed. USA, New York:Addison-Wesley Publishing Company, Inc, 1989

GOLDMAN, A., **Introdução ao escalonamento e aplicações**, Notas de aula: Instituto de matemática e Estatística, IME - USP, 2007. Disponível em: [http://www.ime.usp.br/~gold/cursos/2005/mac5758/notas\\_aula.ps](http://www.ime.usp.br/~gold/cursos/2005/mac5758/notas_aula.ps) . Acesso em 22 out. 2007.

GONÇALVES, J. F., MENDES, J. J. de M., RESENDE, M. G. C., **A hybrid genetic algorithm for the job-shop scheduling problem**. European Journal of Operational Research, 167, p.77-95, 2005.

GOULART, C. P., **Proposta de um modelo de referência para planejamento e controle da produção em empresas virtuais**. Tese (doutorado) – Escola de engenharia, Universidade de São Carlos, São Carlos, 2000.

HERRMANN, J., LEE, C-Y e HINCHMAN, J., **Global job shop scheduling with a genetic algorithm**, Product and operations management, v.4, n.1, 1995.

HEIZER

HOLLAND, J. H., **Adaptation in natural and artificial systems**, Ann Harbor: University of Michigan Press, 1975.

JACKSON, J. R., **scheduling a Production Line to Minimize Maximun Tardiness**, Research Report, University of California at Los Angeles. 1955.

JAIN, A. S., MEERAN, S., **A State-of-the-art Review of Job-Shop Scheduling Techniques**, UK, Scotland: University of Dundee - Department of Applied Physics, Electronic and Mechanical Engineering. 48p. Relatório Técnico. 1998.

JOHNSON, S. M., **Optimal two and three-stage production schedules with setup times included**. Naval Research Logistics Quarterly, Hoboken, v.1, p61-68. 1954.

KIRKPATRICK, S., GELATT, Jr C. D., VECCHI, M. P. **Optimization by Simulated Annealing**, Science, 220, p. 671-680, 1983.  
 LANDMANN, R., **Um modelo heurístico para a programação da produção em fundições com a utilização da lógica fuzzy**. Tese (doutorado), Universidade Federal de Santa Catarina, Florianópolis, 2005.

LINDEN, R., **Algoritmos genéticos**. Brasport, 2 edição. 2008.

LUCERO, A. G. R., **Um método de otimização para a programação da manufatura em pequenos lotes**. 2001. 99f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina. Florianópolis, 2001.

MAYERLE, S. F., **Um algoritmo genético para solução do problema do caixeiro viajante**. Artigo de circulação interna do departamento de engenharia de produção e sistemas da UFSC, 1994.

MATTFELD, D. C. e BIERWIRTH, C., **An efficient algorithm for job shop scheduling with tardiness objectives**. European Journal of Operational Research, 155, p. 616-630, 2004.

MATSUSAKI, C. T. M., **Modelagem de Sistemas de Controle Distribuídos e Colaborativos De Sistemas Produtivos**. Tese (doutorado), Escola Politécnica, Universidade de São Paulo. São Paulo, 2004.

MAURI, G. R. E LORENA, L. A. N. **Simulated Annealing Aplicado a um Modelo Geral do Problema de Roteirização e Programação de Veículos**. XXXVIII SBPO - Simpósio Brasileiro de Pesquisa Operacional. Goiânia, GO - 12 a 15 de setembro de 2006.

MOSCATO, P. **Memetic Algorithms: A Short Introduction**, in New ideas in optimization, p. 219-234, 1 ed. UK, Maidenhead: McGraw-Hill Ltd., UK, 1999.

MITCHEL, M., **An introduction to Genetic Algorithms**. MIT Press, Cambridge, MA. 1996.

MÜLLER, G. I.; GÓMEZ, A. T., **Estratégias de escalonamento em um ambinete de Job-shop**. Anais do XV SEMINCO, p201-208, Blumenau, Brasil, 2006.

OISHI,

PERALES, W., **Classificações dos sistemas de produção**. Anais do Enegep 2001, Salvador, 2001.

PORTER, M. E., **Estratégia Competitiva: Técnicas para análise da indústria e da concorrência**. 7. ed. Rio de Janeiro: Campus, 1991.

PORTER, K, LITTLE, D., PECK, M., ROLLINS, R., **Manufacturing classifications: relationship with production control systems**. Integrated Manufacturing Systems, V.10. N.3-4, p.189-198, 1999.

PINEDO, M., CHAO, X., **Operations Scheduling With Applications in Manufacturing and Services**. Irwing McGraw-Hill, 1999.

RODRIGUES, G.S., **Tecnologia de grupo: uma proposta de arranjo celular para sistemas produtivos**. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Maria, Santa Maria, 2004.

SANTOS FILHO, D. J., **Aspectos do Projeto de Sistemas Produtivos**. Tese (Livre-Docência), Escola Politécnica, Universidade de São Paulo. São Paulo, 2000.

SCOFIELD, W. **Uma implementação híbrida e distribuída do problema Job-shop através de algoritmos Grasp e genético**. Monografia (Projeto Orientado II, Departamento de Computação do Instituto de Ciências Exatas e Biológicas) Universidade Federal de Ouro Preto, Minas Gerais, 2002.

SCHONBERGER, R J., **Técnicas Industriais Japonesas: Nove Lições Ocultas Sobre Simplicidade**. 2. ed. São Paulo: Pioneira, 1984.

SPRECHER, A.; DREXL, A., **Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel**. Kiel, Universität Kiel, 1996.

SLACK, N. et al, **Administração da Produção**, 1 ed. Ed. Atlas. São Paulo, 1997.

SOUZA, M. J. F., **Inteligência Computacional para Otimização**, Notas de aula: Departamento de computação, Instituto de ciências exatas e biológicas - UFOP, 2007. Disponível em: <http://www.decom.ufop.br/prof/marcone/Disciplinas/InteligenciaComputacional/InteligenciaComputacional.pdf> . Acesso em 29 out. 2007.

SPRAKEL, E. B., SEVERIANO FILHO, C., A evolução dos sistemas de PCP sob a ótica da engenharia de produção. In: XIX ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 1999. ABEPRO, Rio de Janeiro. **Anais...** Rio de Janeiro, 1999.

SUBRAMANIAM, V., RAMESH, T., LEE, G. K., WONG, Y. S., HONG, G. S., **Job Shop Scheduling with Dynamic Fuzzy Selection of Dispatching Rules**, International Journal of Advanced Manufacturing Technology 16:759–764, Springer-Verlag London Limited, 2000.

TANOMARU

TAYLOR, S.G., BOLANDER, S.F., **Process Flow Scheduling**, APICS. 1994

TUBINO, D. F., **Manual de Planejamento e controle da Produção**. 2 ed. Ed. Atlas. São Paulo, 2000.

WALTER, C., **Planejamento e Controle da Produção – PCP**. Apostila – Mestrado em Engenharia de Produção – UFRGS, 1999.

WALL, M. B., **A Genetic Algorithm for Resource-Constrained Scheduling**, Tese (doutorado), Department of Mechanical Engineering, Massachusetts Institute of Technology, 1996.

YAMADA, T., **Studies on metaheuristics for job-shop and flow-shop scheduling problems**. Thesis (doctor in informatics: applied mathematics and physics), Kioto University, Kioto, Japan, Nov/2003.

YAMADA, T. e NAKANO R., **A Genetic Algorithm with Multi-Step Crossover for Job-Shop Scheduling Problems**. First IEEE/IEEE International Conference on Genetic ALgorithms in Engineering Systems: Innovations and Applications (GALESIA '95), p. 146-151, Sheffield, UK, 12-14 September 1995.

YAMADA, T. e NAKANO R., **Genetic Algorithms for Job-Shop Scheduling Problems**. Proceedings of Modern Heuristic for Decision Support, p. 67-81, Londres, Março de 1997.

YAMAMOTO, Lia. **Uso de *simulated annealing* e algoritmo genético no problema da reconfiguração de uma rede de distribuição de energia elétrica**, Curitiba. Dissertação (Mestrado) – Ciências Exatas e de Tecnologia, Universidade Federal do Paraná, 2004.

ZHAO, F.; ZHANG, Q.; YANG, Y., **An Improved Particle Swarm Optimization-Based Approach for Production Scheduling Problems**. Proceedings of the 2006 IEEE, International Conference on Mechatronics and Automation. Luoyang, China, 2006.

## APÊNDICE

### A. PROBLEMAS CLÁSSICOS DE AGENDAMENTO JOB-SHOP UTILIZADOS

Os dados aplicados aos testes foram obtidos da literatura, sendo mais precisamente da biblioteca OR-Library de Beasley (2007).

Os dados são tabulados da seguinte forma: cada linha representa uma tarefa ou *job*. As colunas intercalam as operações a executar e o tempo de processamento de cada uma.

Os dados aplicados nos testes origina-se dos seguintes trabalhos:

Abz6	<b>J. Adams, E. Balas and D. Zawack</b> , The shifting bottleneck procedure for job shop scheduling, Management Science 34, 391-401, 1988.
Ft10	<b>H. Fisher, G.L. Thompson</b> , Probabilistic learning combinations of local job-shop scheduling rules, J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, 225-251, 1963.
<b>La02; La21</b>	<b>S. Lawrence</b> , Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.

FONTE: OR-LIBRARY - BEASLEY, J. (2007)

#### TABELA A.1 – FONTES ORIGINAIS DOS TRABALHOS APLICADOS AOS TESTES

A Tabela A.2 a seguir relaciona os valores de *makespan* publicados em diversos trabalhos e organizados a fim de parametrizar comparações.

<i>Problema</i>	<i>Dimensão</i>	<i>Solução ótima</i>
ABZ6	10x10	943
FT10 (MT10)	10x10	930
LA02	10x5	655
LA21	15X10	1046

FONTE: GONÇALVES, MENDES E RESENDE (2005); EL-BOURI ET AL (2007) E SCHULTZ ET AL (2007)

#### TABELA A.2 – VALORES ÓTIMOS DOS PROBLEMAS UTILIZADOS NOS TESTES

A.1. Problema ABZ6<sub>[10 X 10]</sub>

7	62	8	24	5	25	3	84	4	47	6	38	2	82	0	93	9	24	1	66
5	47	2	97	8	92	9	22	1	93	4	29	7	56	3	80	0	78	6	67
1	45	7	46	6	22	2	26	9	38	0	69	4	40	3	33	8	75	5	96
4	85	8	76	5	68	9	88	3	36	6	75	2	56	1	35	0	77	7	85
8	60	9	20	7	25	3	63	4	81	0	52	1	30	5	98	6	54	2	86
3	87	9	73	5	51	2	95	4	65	1	86	6	22	8	58	0	80	7	65
5	81	2	53	7	57	6	71	9	81	0	43	4	26	8	54	3	58	1	69
4	20	6	86	5	21	8	79	9	62	2	34	0	27	1	81	7	30	3	46
9	68	6	66	5	98	8	86	7	66	0	56	3	82	1	95	4	47	2	78
0	30	3	50	7	34	2	58	1	77	5	34	8	84	4	40	9	46	6	44

FONTE: BEASLEY, J. (2007)

A.2. Problema LA02<sub>[10 X 5]</sub>

0	20	3	87	1	31	4	76	2	17
4	25	2	32	0	24	1	18	3	81
1	72	2	23	4	28	0	58	3	99
2	86	1	76	4	97	0	45	3	90
4	27	0	42	3	48	2	17	1	46
1	67	0	98	4	48	3	27	2	62
4	28	1	12	3	19	0	80	2	50
1	63	0	94	2	98	3	50	4	80
4	14	0	75	2	50	1	41	3	55
4	72	2	18	1	37	3	79	0	61

FONTE: BEASLEY, J. (2007)

A.3. Problema LA21<sub>[15 X 10]</sub>

2	34	3	55	5	95	9	16	4	21	6	71	0	53	8	52	1	21	7	26
3	39	2	31	0	12	1	42	9	79	8	77	6	77	5	98	4	55	7	66
1	19	0	83	3	34	4	92	6	54	9	79	8	62	5	37	2	64	7	43
4	60	2	87	8	24	5	77	3	69	7	38	1	87	6	41	9	83	0	93
8	79	9	77	2	98	4	96	3	17	0	44	7	43	6	75	1	49	5	25
8	35	7	95	6	9	9	10	2	35	1	7	5	28	4	61	0	95	3	76
4	28	5	59	3	16	9	43	0	46	8	50	6	52	7	27	2	59	1	91
5	9	4	20	2	39	6	54	1	45	7	71	0	87	3	41	9	43	8	14
1	28	5	33	0	78	3	26	2	37	7	8	8	66	6	89	9	42	4	33
2	94	5	84	6	78	9	81	1	74	3	27	8	69	0	69	7	45	4	96
1	31	4	24	0	20	2	17	9	25	8	81	5	76	3	87	7	32	6	18
5	28	9	97	0	58	4	45	6	76	3	99	2	23	1	72	8	90	7	86
5	27	9	48	8	27	7	62	4	98	6	67	3	48	0	42	1	46	2	17
1	12	8	50	0	80	2	50	9	80	3	19	5	28	6	63	4	94	7	98
4	61	3	55	6	37	5	14	2	50	8	79	1	41	9	72	7	18	0	75

FONTE: BEASLEY, J. (2007)

A.4. Problema MT10<sub>[10 x 10]</sub> (ou FT100)

0	29	1	78	2	9	3	36	4	49	5	11	6	62	7	56	8	44	9	21
0	43	2	90	4	75	9	11	3	69	1	28	6	46	5	46	7	72	8	30
1	91	0	85	3	39	2	74	8	90	5	10	7	12	6	89	9	45	4	33
1	81	2	95	0	71	4	99	6	9	8	52	7	85	3	98	9	22	5	43
2	14	0	6	1	22	5	61	3	26	4	69	8	21	7	49	9	72	6	53
2	84	1	2	5	52	3	95	8	48	9	72	0	47	6	65	4	6	7	25
1	46	0	37	3	61	2	13	6	32	5	21	9	32	8	89	7	30	4	55
2	31	0	86	1	46	5	74	4	32	6	88	8	19	9	48	7	36	3	79
0	76	1	69	3	76	5	51	2	85	9	11	6	40	7	89	4	26	8	74
1	85	0	13	2	61	6	7	8	64	9	76	5	47	3	52	4	90	7	45



B. DADOS DA SOLUÇÃO DO PROBLEMA CLÁSSICO ABZ6

Dados								
Problema:	ABZ6	SPI						
Dimensão:	10X10	SPRT						
Populações:	100	SPTq						
	50	RHC						
Taxas Mut.	5%	RND						
	1%							
L.Bound	943							
Resumo								
	Tempo	Média	E%med	Variância		Tempo	Menor	E%min
SPTI_50_A_1	00:00,163	1097,00	16,3%	0,00		00:00,125	1097,00	16,3%
SPRT_50_A_1	02:48,956	1064,20	12,9%	85,20		00:36,720	1055,00	11,9%
SPTq_50_A_1	00:00,237	1097,00	16,3%	0,00		00:00,110	1097,00	16,3%
RHC_50_A_1	02:27,516	1083,00	14,8%	76,50		00:00,139	1073,00	13,8%
RND_50_A_1	02:09,324	1046,40	11,0%	73,30		00:01,624	1037,00	10,0%
SPTI_50_ND_1	00:00,593	979,80	3,9%	0,20		00:00,204	979,00	3,8%
SPRT_50_ND_1	01:43,404	971,00	3,0%	0,00		00:48,772	971,00	3,0%
SPTq_50_ND_1	01:04,541	978,20	3,7%	3,20		00:35,221	975,00	3,4%
RHC_50_ND_1	01:19,264	976,20	3,5%	14,70		00:00,499	971,00	3,0%
RND_50_ND_1	01:45,775	971,00	3,0%	0,00		00:04,132	971,00	3,0%
SPTI_100_A_1	00:00,210	1097,00	16,3%	0,00		00:00,157	1097,00	16,3%
SPRT_100_A_1	03:41,082	1037,60	10,0%	53,30		00:38,227	1030,00	9,2%
SPTq_100_A_1	00:00,253	1097,00	16,3%	0,00		00:00,172	1097,00	16,3%
RHC_100_A_1	01:53,491	1073,40	13,8%	0,80		00:00,062	1072,00	13,7%
RND_100_A_1	04:39,970	1048,40	11,2%	31,80		00:39,656	1043,00	10,6%
SPTI_100_ND_1	02:53,008	975,40	3,4%	8,80		00:00,079	971,00	3,0%
SPRT_100_ND_1	00:55,987	971,00	3,0%	0,00		00:05,922	971,00	3,0%
SPTq_100_ND_1	04:24,313	979,20	3,8%	0,20		00:26,467	979,00	3,8%
RHC_100_ND_1	02:49,951	971,80	3,1%	3,20		01:15,330	971,00	3,0%
RND_100_ND_1	03:13,887	971,80	3,1%	3,20		00:05,140	971,00	3,0%
SPTI_50_A_5	00:00,142	1097,00	16,3%	0,00		00:00,108	1097,00	16,3%
SPRT_50_A_5	01:38,564	1063,20	12,7%	93,70		00:04,727	1051,00	11,5%
SPTq_50_A_5	00:00,121	1097,00	16,3%	0,00		00:00,109	1097,00	16,3%
RHC_50_A_5	01:14,225	1080,60	14,6%	86,80		00:39,058	1069,00	13,4%
RND_50_A_5	01:12,763	1047,40	11,1%	38,30		00:33,260	1042,00	10,5%
SPTI_50_ND_5	00:00,378	979,80	3,9%	0,20		00:00,110	979,00	3,8%
SPRT_50_ND_5	00:27,896	971,80	3,1%	3,20		00:03,568	971,00	3,0%
SPTq_50_ND_5	00:44,539	979,20	3,8%	0,20		00:02,249	979,00	3,8%
RHC_50_ND_5	00:30,608	979,00	3,8%	5,00		00:04,560	975,00	3,4%
RND_50_ND_5	01:24,677	971,00	3,0%	0,00		00:18,994	971,00	3,0%
SPTI_100_A_5	00:00,238	1097,00	16,3%	0,00		00:00,172	1097,00	16,3%
SPRT_100_A_5	02:59,027	1053,60	11,7%	57,80		00:06,656	1043,00	10,6%
SPTq_100_A_5	00:00,241	1097,00	16,3%	0,00		00:00,156	1097,00	16,3%
RHC_100_A_5	02:57,282	1071,20	13,6%	22,70		00:00,063	1063,00	12,7%
RND_100_A_5	06:14,118	1031,40	9,4%	94,80		02:49,828	1016,00	7,7%
SPTI_100_ND_5	02:19,824	976,20	3,5%	1,20		00:06,312	975,00	3,4%
SPRT_100_ND_5	00:53,491	971,00	3,0%	0,00		00:04,894	971,00	3,0%
SPTq_100_ND_5	05:13,304	978,20	3,7%	3,20		01:57,518	975,00	3,4%
RHC_100_ND_5	02:50,627	975,40	3,4%	20,30		00:01,094	971,00	3,0%
RND_100_ND_5	02:23,024	971,80	3,1%	3,20		00:26,641	971,00	3,0%
Menor	06:14,118	971	3,0%				971	3,0%

FONTE: PRÓPRIA

QUADRO B.I – RESUMO GERAL DA SOLUÇÃO DO PROBLEMA ABZ6: DADOS MÉDIOS E MÍNIMOS

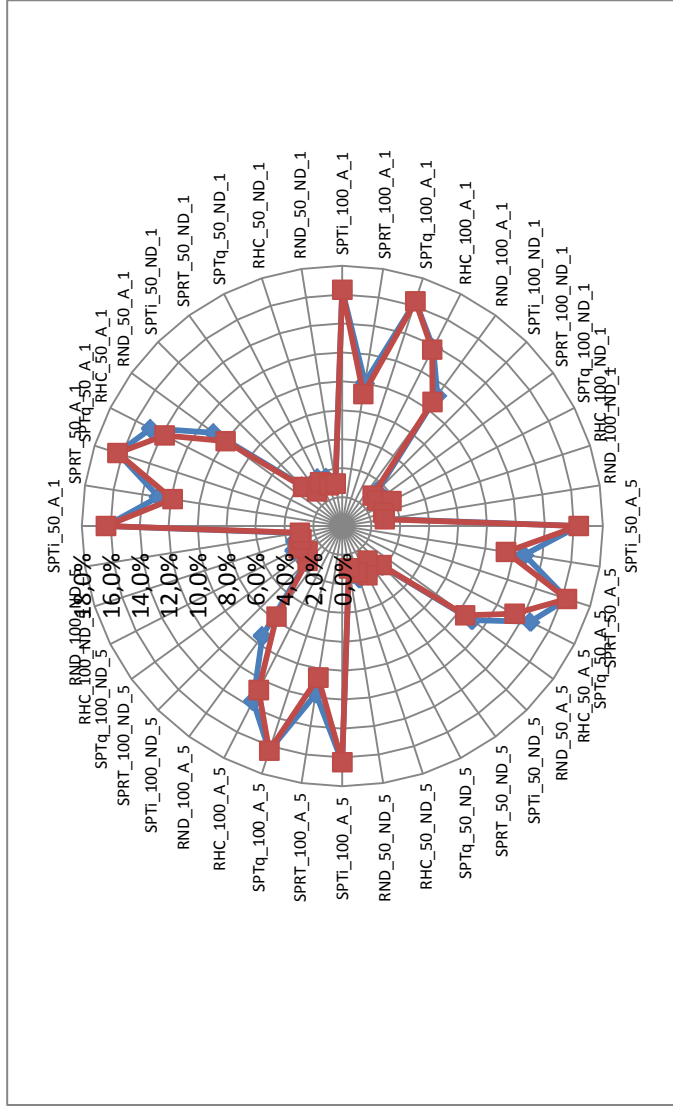


FIGURA B.1 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

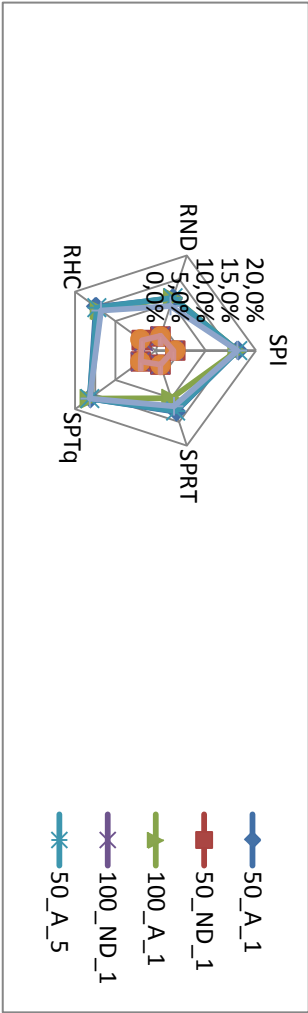


FIGURA B.2 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

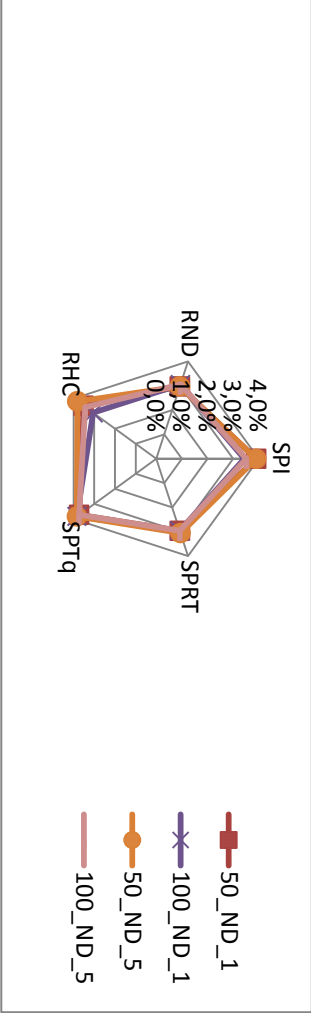


FIGURA B.3 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

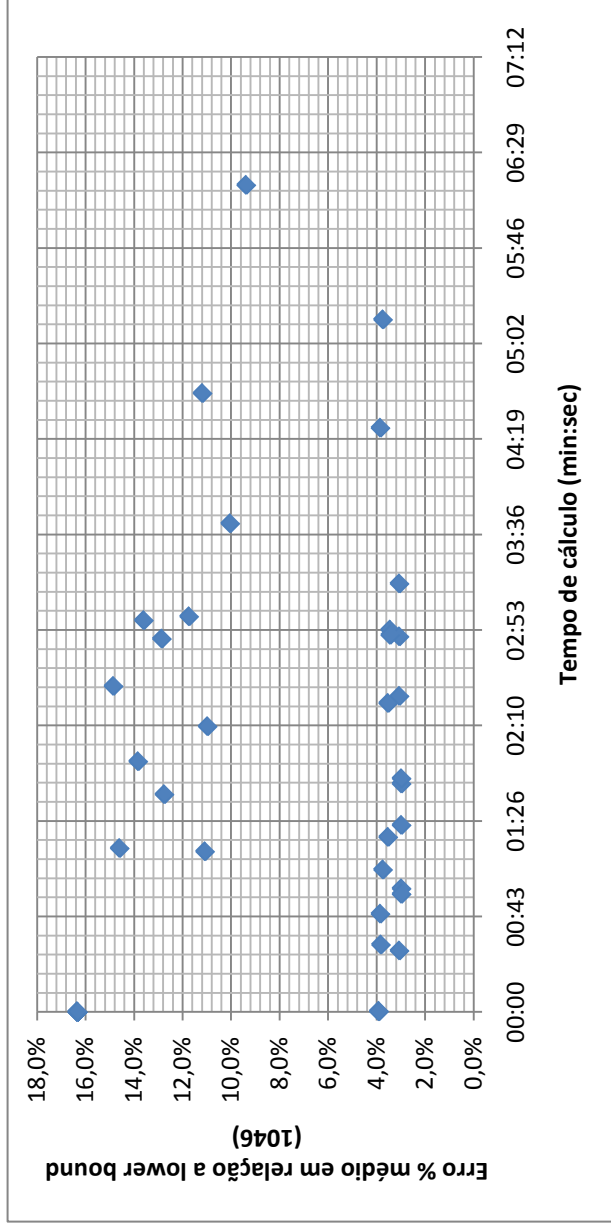


FIGURA B.4 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

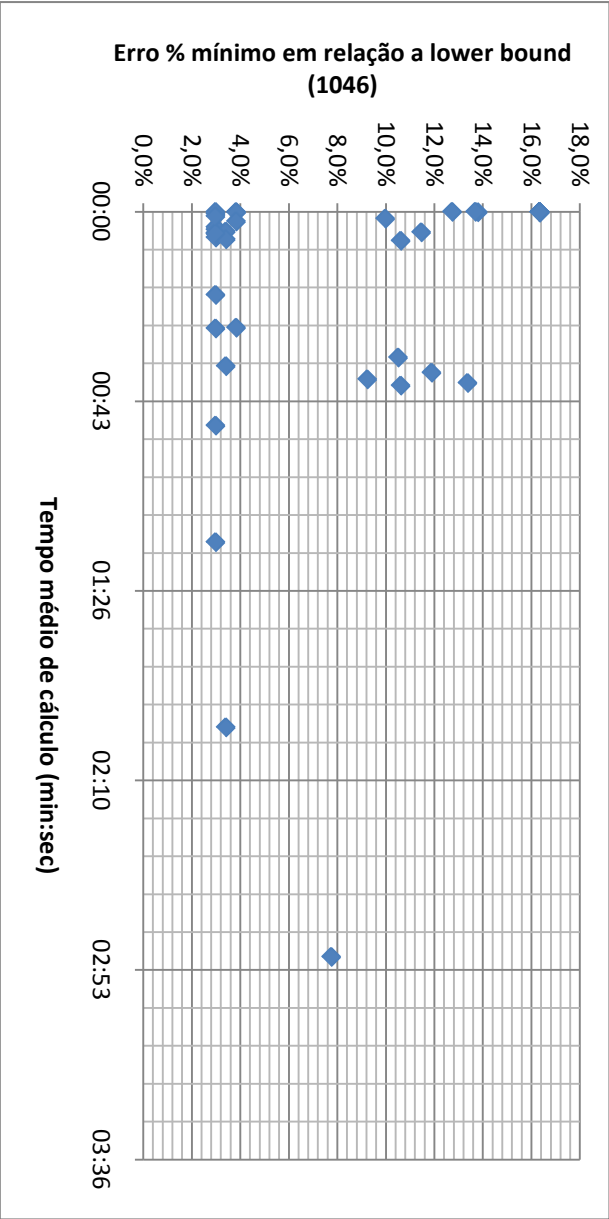


FIGURA B.5 – GRÁFICO RELACIONANDO ERRO % MÍNIMO EM RELAÇÃO AO ÓTIMO.

C. DADOS DA SOLUÇÃO DO PROBLEMA CLÁSSICO LA02

Dados								
Problema:	LA02	SPI						
Dimensão:	10X5	SPRT						
Populações:	160	SPTq						
	80	RHC						
Taxas Mut.	5%	RND						
	1%							
L.Bound	655							
Resumo								
	Tempo	Média	E%med	Variação		Tempo	Menor	E%min
SPTi_50_A_1	00:56,202	753,20	15,0%	67,70		00:00,032	745,00	13,7%
SPRT_50_A_1	01:25,464	745,40	13,8%	19,80		00:01,032	738,00	12,7%
SPTq_50_A_1	01:22,138	739,40	12,9%	34,30		00:07,785	735,00	12,2%
RHC_50_A_1	01:37,572	735,60	12,3%	14,30		00:06,109	730,00	11,5%
RND_50_A_1	00:38,426	724,20	10,6%	42,20		00:06,439	714,00	9,0%
SPTi_50_ND_1	00:56,350	681,80	4,1%	19,20		00:09,419	677,00	3,4%
SPRT_50_ND_1	01:18,300	684,60	4,5%	24,30		00:29,466	676,00	3,2%
SPTq_50_ND_1	01:31,525	684,80	4,5%	0,20		00:34,610	684,00	4,4%
RHC_50_ND_1	00:45,550	688,20	5,1%	16,70		00:10,967	685,00	4,6%
RND_50_ND_1	01:07,590	676,60	3,3%	15,30		00:38,692	672,00	2,6%
SPTi_100_A_1	01:09,293	740,20	13,0%	69,70		00:00,952	729,00	11,3%
SPRT_100_A_1	02:39,551	739,40	12,9%	42,80		00:34,265	730,00	11,5%
SPTq_100_A_1	03:18,256	716,00	9,3%	56,00		01:38,517	709,00	8,2%
RHC_100_A_1	01:09,449	727,20	11,0%	31,70		00:02,578	719,00	9,8%
RND_100_A_1	02:49,073	718,40	9,7%	40,30		00:47,436	712,00	8,7%
SPTi_100_ND_1	01:09,487	677,00	3,4%	0,00		00:01,672	677,00	3,4%
SPRT_100_ND_1	02:35,561	682,40	4,2%	35,30		00:17,295	676,00	3,2%
SPTq_100_ND_1	03:35,297	682,00	4,1%	3,00		00:10,454	681,00	4,0%
RHC_100_ND_1	01:59,534	686,80	4,9%	2,70		00:00,891	685,00	4,6%
RND_100_ND_1	02:58,667	673,40	2,8%	9,80		01:00,494	670,00	2,3%
SPTi_50_A_5	01:06,456	752,40	14,9%	42,80		00:00,140	744,00	13,6%
SPRT_50_A_5	01:43,907	747,40	14,1%	30,80		00:27,670	738,00	12,7%
SPTq_50_A_5	01:52,491	736,60	12,5%	55,80		00:01,641	727,00	11,0%
RHC_50_A_5	00:53,486	736,40	12,4%	10,30		00:10,014	733,00	11,9%
RND_50_A_5	01:49,275	712,20	8,7%	67,70		00:01,062	705,00	7,6%
SPTi_50_ND_5	01:48,959	679,40	3,7%	12,80		00:02,484	677,00	3,4%
SPRT_50_ND_5	00:57,375	686,60	4,8%	42,80		00:15,327	676,00	3,2%
SPTq_50_ND_5	01:32,640	684,40	4,5%	1,80		00:16,203	682,00	4,1%
RHC_50_ND_5	00:23,712	692,40	5,7%	31,30		00:12,455	687,00	4,9%
RND_50_ND_5	00:33,342	678,40	3,6%	10,80		00:07,281	676,00	3,2%
SPTi_100_A_5	02:47,648	732,20	11,8%	27,20		01:10,828	728,00	11,1%
SPRT_100_A_5	03:29,966	742,80	13,4%	13,70		00:59,968	739,00	12,8%
SPTq_100_A_5	01:38,915	707,80	8,1%	2,70		00:01,611	706,00	7,8%
RHC_100_A_5	02:09,070	722,40	10,3%	41,30		00:28,299	714,00	9,0%
RND_100_A_5	01:48,775	695,50	6,2%	65,90		01:19,687	686,00	4,7%
SPTi_100_ND_5	02:13,561	677,00	3,4%	0,00		00:01,718	677,00	3,4%
SPRT_100_ND_5	02:20,664	684,60	4,5%	24,30		00:02,047	676,00	3,2%
SPTq_100_ND_5	02:18,720	682,20	4,2%	2,70		01:14,889	681,00	4,0%
RHC_100_ND_5	01:28,230	684,40	4,5%	14,30		00:29,531	678,00	3,5%
RND_100_ND_5	02:44,860	675,80	3,2%	0,20		00:13,639	675,00	3,1%
Menor	03:35,297	673,4	2,8%				670	2,3%

FONTE: PRÓPRIA  
QUADRO C.1 – RESUMO GERAL DA SOLUÇÃO DO PROBLEMA LA02:  
DADOS MÉDIOS E MÍNIMOS

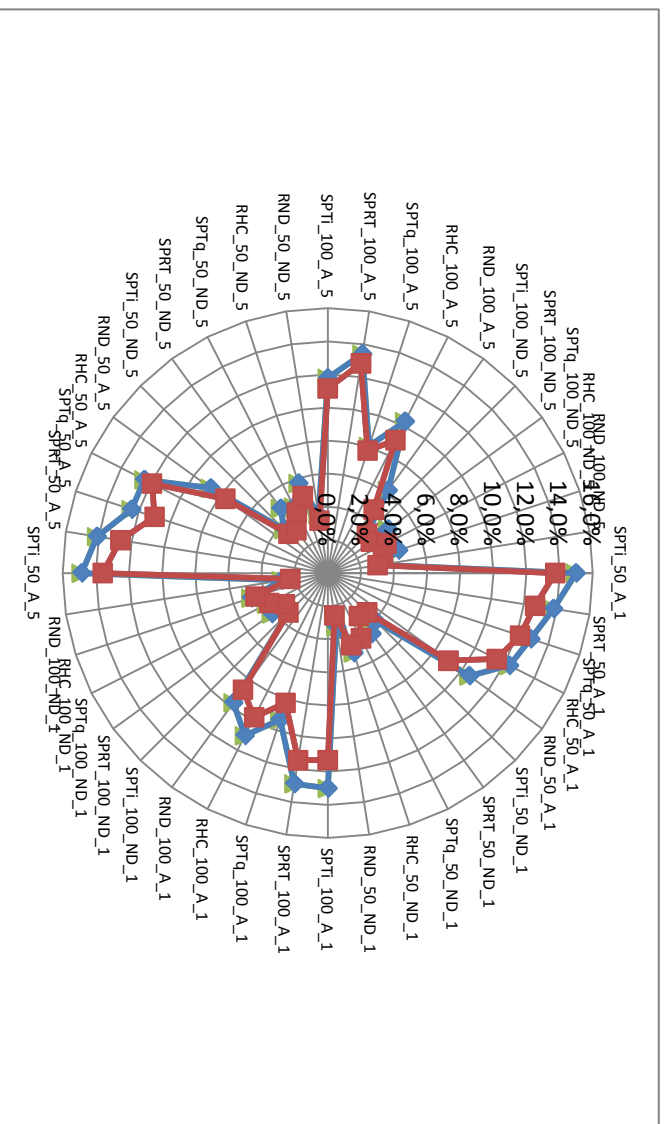


FIGURA C.1 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

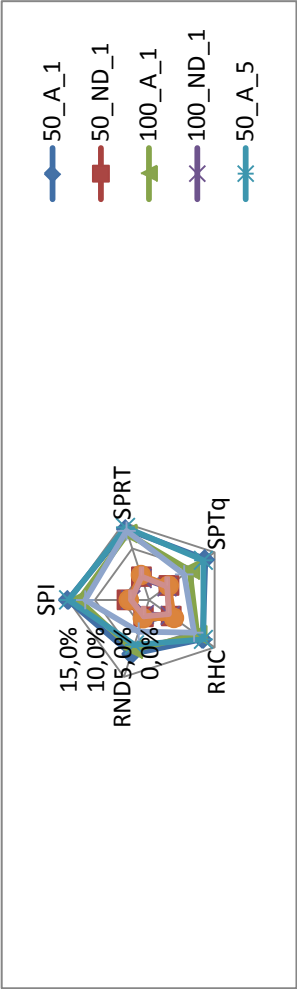


FIGURA C.2 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

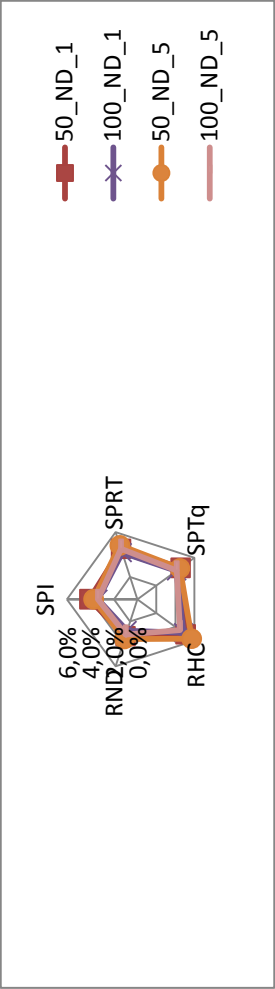


FIGURA C.3 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.



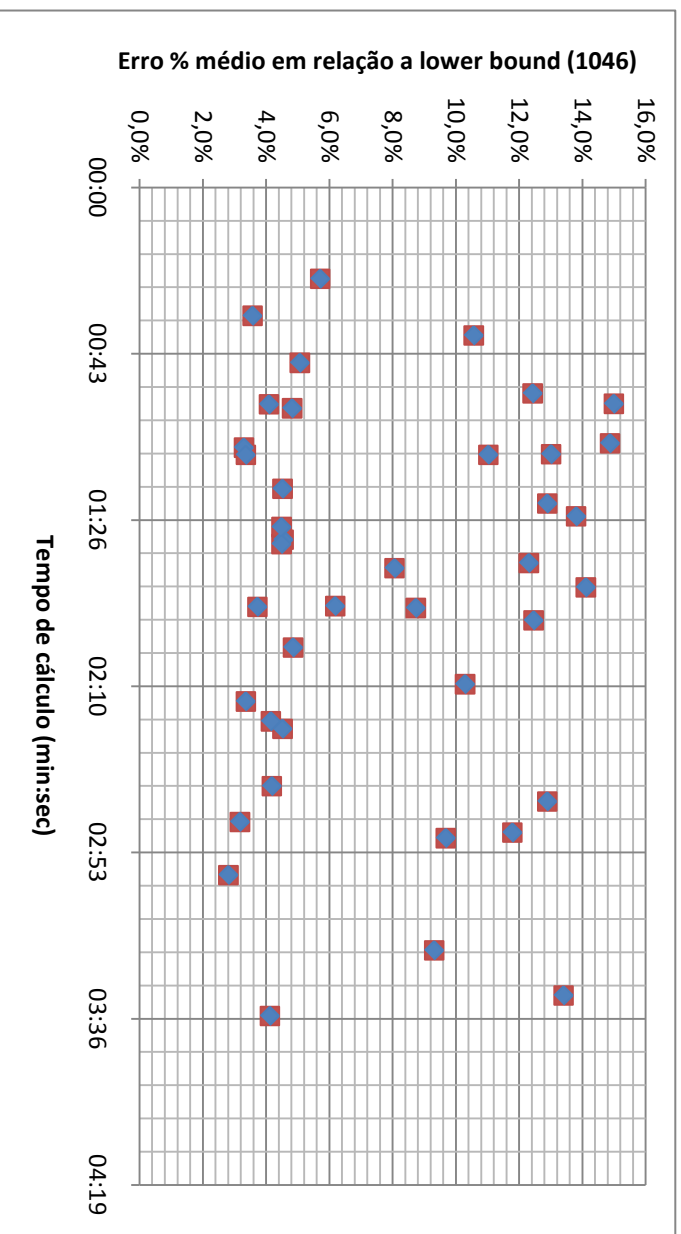


FIGURA C.4 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

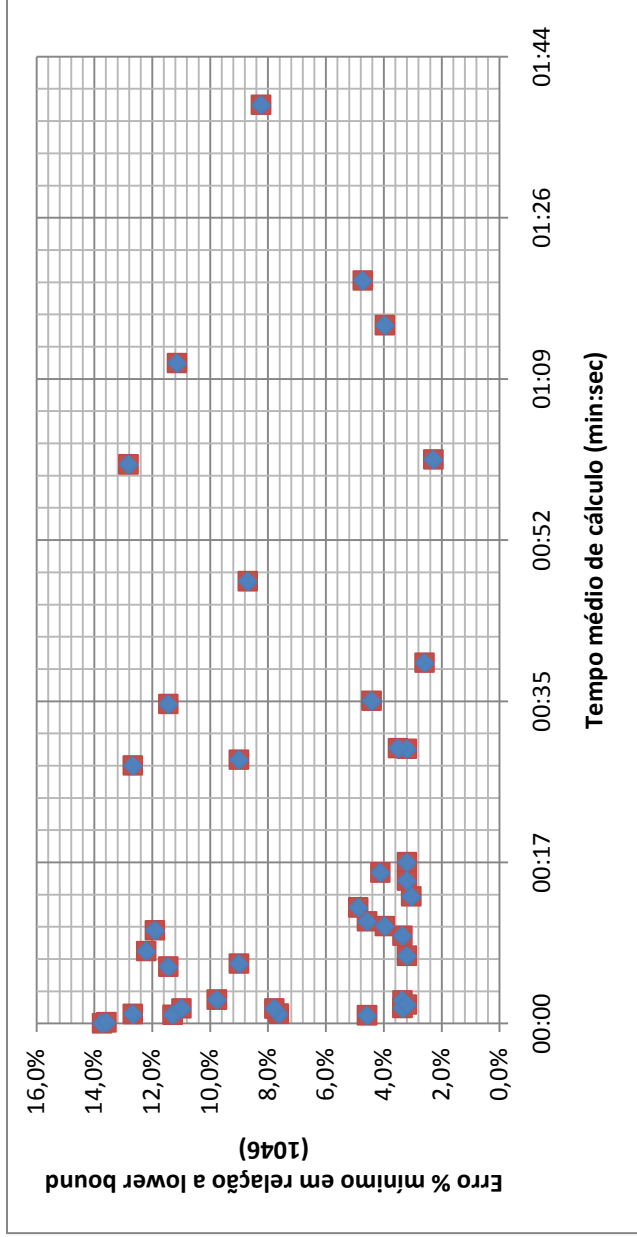


FIGURA C.5 – GRÁFICO RELACIONANDO ERRO % MÍNIMO EM RELAÇÃO AO ÓTIMO.

D. Dados da solução do problema clássico LA21

Dados								
Problema:	LA21	SPI						
Dimensão:	15x10	SPRT						
Populações:	160	SPTq						
	80	RHC						
Taxas Mut.	5%	RND						
	1%							
L.Bound	1046							
Resumo								
	Tempo	Média	E%med	Variância	Tempo	Menor	E%med	
SPTI_80_A_1	00:00,422	1313,00	25,5%	0,00	00:00,297	1313,00	25,5%	
SPRT_80_A_1	00:00,409	1313,00	25,5%	0,00	00:00,297	1313,00	25,5%	
SPTq_80_A_1	00:00,422	1313,00	25,5%	0,00	00:00,312	1313,00	25,5%	
RHC_80_A_1	03:13,061	1309,33	25,2%	105,50	00:00,344	1282,00	22,6%	
RND_80_A_1	05:33,647	1241,00	18,6%	59,00	01:53,393	1230,00	17,6%	
SPTI_80_ND_1	12:19,668	1152,60	10,2%	61,30	03:49,354	1146,00	9,6%	
SPRT_80_ND_1	09:05,707	1179,60	12,8%	29,80	00:46,640	1173,00	12,1%	
SPTq_80_ND_1	03:42,105	1179,20	12,7%	89,20	01:03,437	1165,00	11,4%	
RHC_80_ND_1	04:47,656	1164,40	11,3%	80,30	00:19,436	1152,00	10,1%	
RND_80_ND_1	12:42,130	1127,80	7,8%	55,20	07:17,521	1120,00	7,1%	
SPTI_160_A_1	00:47,519	1312,40	25,5%	1,80	00:00,610	1310,00	25,2%	
SPRT_160_A_1	00:00,675	1313,00	25,5%	0,00	00:00,609	1313,00	25,5%	
SPTq_160_A_1	00:00,757	1313,00	25,5%	0,00	00:00,610	1313,00	25,5%	
RHC_160_A_1	12:16,305	1308,20	25,1%	37,70	00:01,172	1298,00	24,1%	
RND_160_A_1	19:13,046	1243,40	18,9%	85,80	01:00,653	1230,00	17,6%	
SPTI_160_ND_1	24:56,157	1148,40	9,8%	44,80	05:19,031	1139,00	8,9%	
SPRT_160_ND_1	07:58,764	1145,80	9,5%	141,70	00:25,782	1130,00	8,0%	
SPTq_160_ND_1	13:20,103	1161,40	11,0%	35,30	09:18,232	1152,00	10,1%	
RHC_160_ND_1	17:03,277	1154,00	10,3%	36,50	06:51,687	1147,00	9,7%	
RND_160_ND_1	24:28,807	1120,40	7,1%	76,80	12:31,124	1107,00	5,8%	
SPTI_80_A_5	00:00,338	1313,00	25,5%	0,00	00:00,313	1313,00	25,5%	
SPRT_80_A_5	00:00,391	1313,00	25,5%	0,00	00:00,296	1313,00	25,5%	
SPTq_80_A_5	00:00,294	1313,00	25,5%	0,00	00:00,281	1313,00	25,5%	
RHC_80_A_5	02:14,120	1308,40	25,1%	67,80	00:00,328	1294,00	23,7%	
RND_80_A_5	06:49,432	1249,80	19,5%	62,20	02:05,563	1236,00	18,2%	
SPTI_80_ND_5	03:55,129	1156,40	10,6%	27,30	00:58,189	1148,00	9,8%	
SPRT_80_ND_5	09:25,361	1175,20	12,4%	31,70	07:07,460	1168,00	11,7%	
SPTq_80_ND_5	05:19,787	1162,20	11,1%	101,20	01:52,138	1146,00	9,6%	
RHC_80_ND_5	06:12,307	1158,00	10,7%	98,50	01:47,530	1147,00	9,7%	
RND_80_ND_5	06:00,899	1122,20	7,3%	24,20	00:37,601	1114,00	6,5%	
SPTI_160_A_5	00:35,653	1311,40	25,4%	12,80	00:00,719	1305,00	24,8%	
SPRT_160_A_5	00:00,931	1313,00	25,5%	0,00	00:00,704	1313,00	25,5%	
SPTq_160_A_5	00:01,000	1313,00	25,5%	0,00	00:00,609	1313,00	25,5%	
RHC_160_A_5	05:28,141	1311,40	25,4%	12,80	00:00,671	1305,00	24,8%	
RND_160_A_5	22:33,957	1240,40	18,6%	45,30	05:24,767	1234,00	18,0%	
SPTI_160_ND_5	27:51,973	1146,80	9,6%	25,70	13:12,058	1139,00	8,9%	
SPRT_160_ND_5	20:52,328	1168,80	11,7%	25,20	02:24,721	1161,00	11,0%	
SPTq_160_ND_5	10:47,182	1144,20	9,4%	148,70	01:28,544	1125,00	7,6%	
RHC_160_ND_5	13:27,977	1144,00	9,4%	71,50	07:32,965	1132,00	8,2%	
RND_160_ND_5	16:09,744	1124,80	7,5%	40,70	02:26,998	1116,00	6,7%	
Menor	27:51,973	1120,4	7,1%			1107	5,8%	

FONTE: PRÓPRIA

QUADRO D.1 – RESUMO GERAL DA SOLUÇÃO DO PROBLEMA LA21: DADOS MÉDIOS E MÍNIMOS

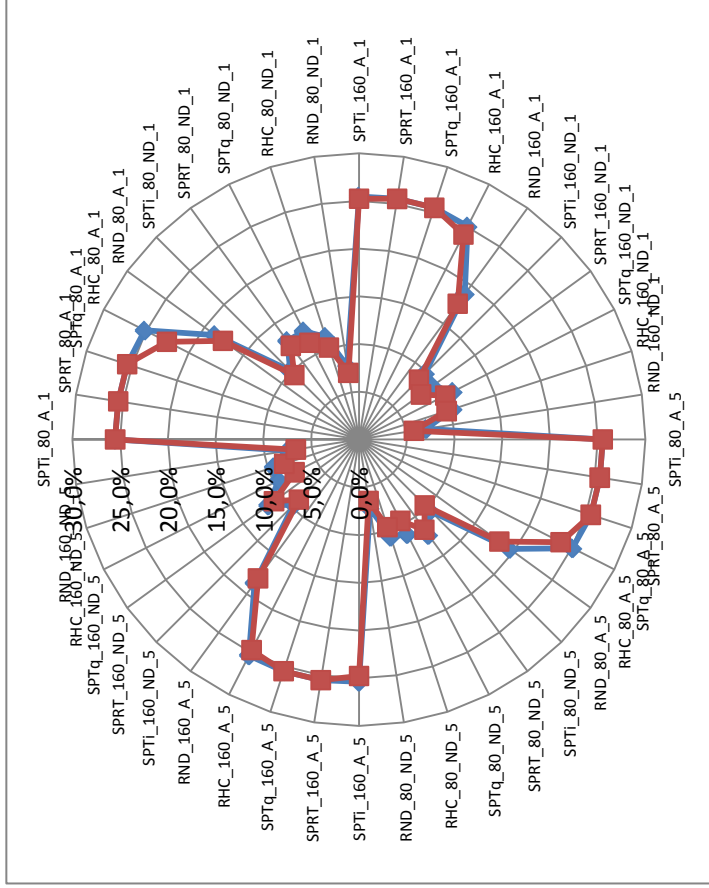


FIGURA D.1 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

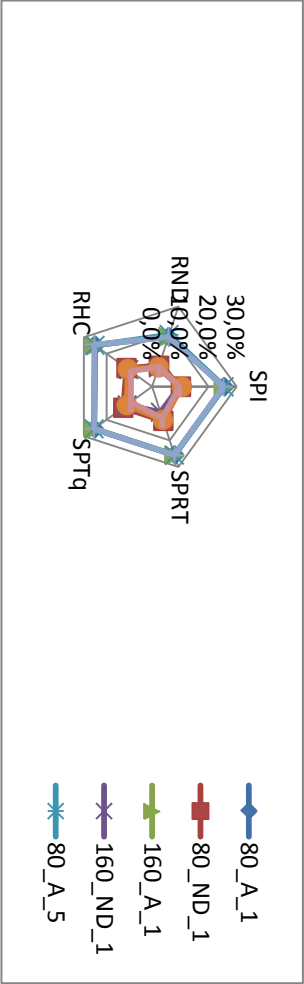


FIGURA D.2 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

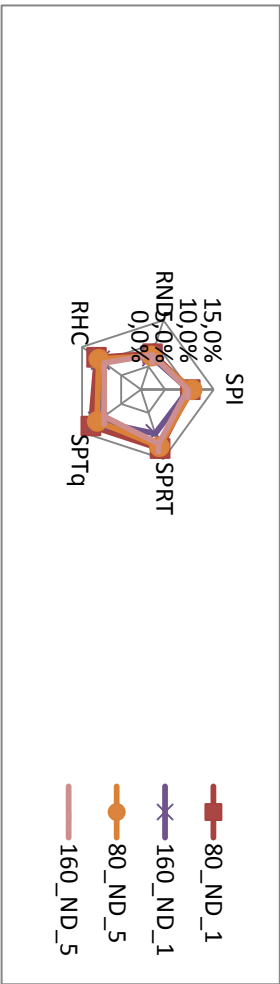


FIGURA D.3 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

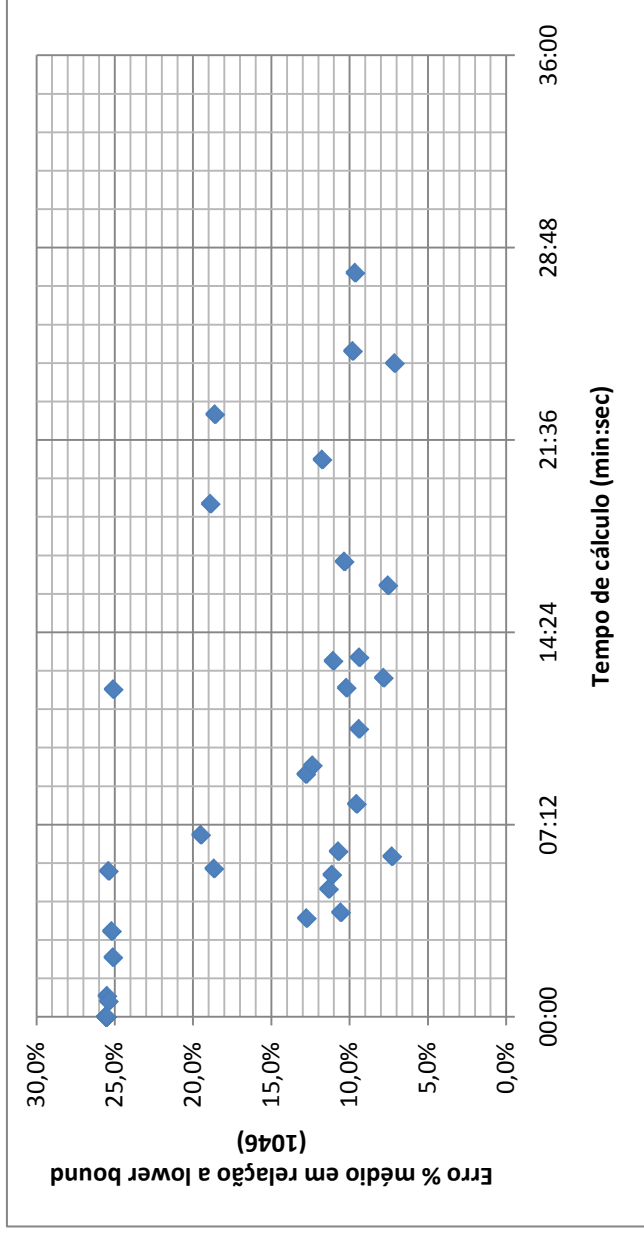


FIGURA D.4 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

FIGURA D.5 – GRÁFICO RELACIONANDO ERRO % MÍNIMO EM RELAÇÃO AO ÓTIMO.

E. DADOS DA SOLUÇÃO DO PROBLEMA CLÁSSICO FT10

Dados								
Problema:	FT10	SPI						
Dimensão:	10X10	SPRT						
Populações:	100	SPTq						
	50	RHC						
Taxas Mut.	5%	RND						
	1%							
L.Bound	930							
Resumo								
	Tempo	Média	E%med	Variância	Tempo	Menor	E%min	
SPTi_50_A_1	00:00,209	1074,00	15,5%	0,00	00:00,093	1074,00	15,5%	
SPRT_50_A_1	00:00,128	1074,00	15,5%	0,00	00:00,093	1074,00	15,5%	
SPTq_50_A_1	00:00,119	1074,00	15,5%	0,00	00:00,079	1074,00	15,5%	
RHC_50_A_1	00:00,140	1074,00	15,5%	0,00	00:00,078	1074,00	15,5%	
RND_50_A_1	00:00,106	1074,00	15,5%	0,00	00:00,078	1074,00	15,5%	
SPTi_50_ND_1	01:57,180	988,80	6,3%	50,70	00:01,985	978,00	5,2%	
SPRT_50_ND_1	02:45,311	1018,60	9,5%	79,30	01:34,720	1003,00	7,8%	
SPTq_50_ND_1	03:28,699	1014,20	9,1%	7,70	02:30,206	1010,00	8,6%	
RHC_50_ND_1	02:05,665	994,40	6,9%	17,80	00:11,642	989,00	6,3%	
RND_50_ND_1	02:37,427	1002,60	7,8%	47,30	00:53,327	993,00	6,8%	
SPTi_100_A_1	00:00,225	1074,00	15,5%	0,00	00:00,172	1074,00	15,5%	
SPRT_100_A_1	00:00,184	1074,00	15,5%	0,00	00:00,172	1074,00	15,5%	
SPTq_100_A_1	00:00,200	1074,00	15,5%	0,00	00:00,171	1074,00	15,5%	
RHC_100_A_1	00:11,513	1071,40	15,2%	33,80	00:00,188	1061,00	14,1%	
RND_100_A_1	02:39,800	1071,60	15,2%	14,30	00:00,172	1065,00	14,5%	
SPTi_100_ND_1	01:43,327	988,60	6,3%	8,30	00:01,765	984,00	5,8%	
SPRT_100_ND_1	04:45,028	1021,60	9,8%	59,80	01:45,193	1013,00	8,9%	
SPTq_100_ND_1	03:21,633	1010,60	8,7%	1,80	01:08,513	1010,00	8,6%	
RHC_100_ND_1	03:37,271	992,60	6,7%	3,30	00:05,876	990,00	6,5%	
RND_100_ND_1	03:17,916	984,00	5,8%	51,50	00:04,032	976,00	4,9%	
SPTi_50_A_5	00:00,100	1074,00	15,5%	0,00	00:00,078	1074,00	15,5%	
SPRT_50_A_5	00:00,100	1074,00	15,5%	0,00	00:00,078	1074,00	15,5%	
SPTq_50_A_5	00:00,207	1074,00	15,5%	0,00	00:00,078	1074,00	15,5%	
RHC_50_A_5	00:00,156	1074,00	15,5%	0,00	00:00,077	1074,00	15,5%	
RND_50_A_5	01:18,218	1073,20	15,4%	3,20	00:00,093	1070,00	15,1%	
SPTi_50_ND_5	00:35,914	995,60	7,1%	18,80	00:01,174	989,00	6,3%	
SPRT_50_ND_5	02:36,586	1035,60	11,4%	30,30	00:01,562	1028,00	10,5%	
SPTq_50_ND_5	03:40,617	1013,20	8,9%	8,20	00:51,830	1010,00	8,6%	
RHC_50_ND_5	01:58,026	995,00	7,0%	4,00	00:30,237	993,00	6,8%	
RND_50_ND_5	03:38,589	994,40	6,9%	12,80	02:03,261	990,00	6,5%	
SPTi_100_A_5	00:00,319	1074,00	15,5%	0,00	00:00,172	1074,00	15,5%	
SPRT_100_A_5	00:00,191	1074,00	15,5%	0,00	00:00,171	1074,00	15,5%	
SPTq_100_A_5	00:00,334	1074,00	15,5%	0,00	00:00,171	1074,00	15,5%	
RHC_100_A_5	00:00,259	1074,00	15,5%	0,00	00:00,171	1074,00	15,5%	
RND_100_A_5	00:00,303	1074,00	15,5%	0,00	00:00,187	1074,00	15,5%	
SPTi_100_ND_5	03:56,858	986,20	6,0%	15,20	00:25,049	981,00	5,5%	
SPRT_100_ND_5	05:43,376	1020,40	9,7%	45,30	01:12,651	1012,00	8,8%	
SPTq_100_ND_5	06:25,507	1010,40	8,6%	18,30	00:44,031	1005,00	8,1%	
RHC_100_ND_5	04:35,202	993,40	6,8%	2,30	01:30,082	992,00	6,7%	
RND_100_ND_5	05:39,659	990,00	6,5%	14,50	01:07,437	984,00	5,8%	
Menor	06:25,507	984	5,8%			976	4,9%	

FONTE: PRÓPRIA  
QUADRO E.1 – RESUMO GERAL DA SOLUÇÃO DO PROBLEMA FT10:  
DADOS MÉDIOS E MÍNIMOS



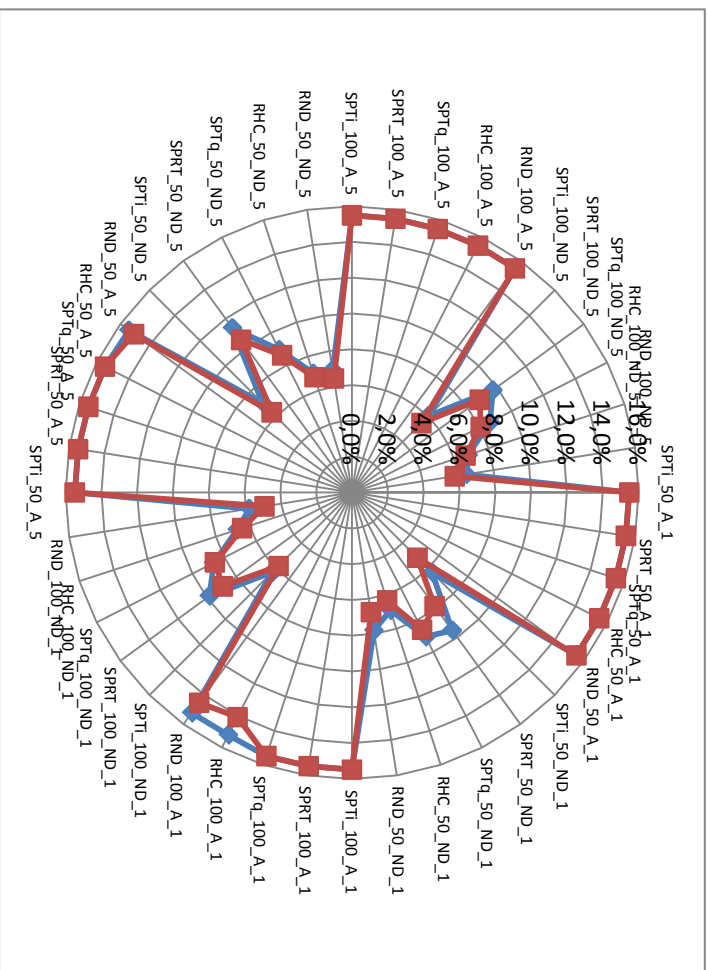


FIGURA E.1 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

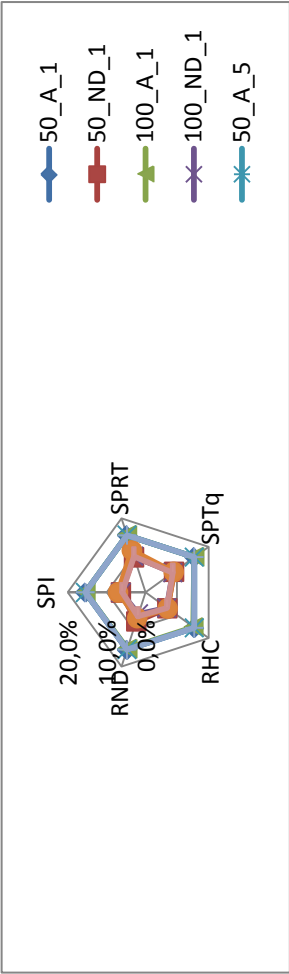


FIGURA E.2 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

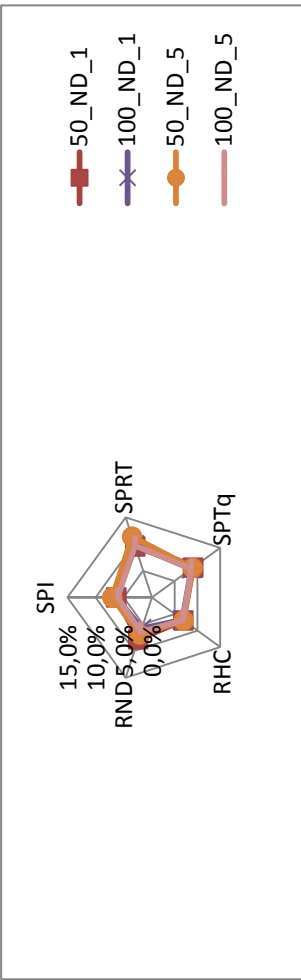


FIGURA E.3 – GRÁFICO RELACIONANDO ERRO % MÉDIO E MÍNIMO EM RELAÇÃO AO ÓTIMO.

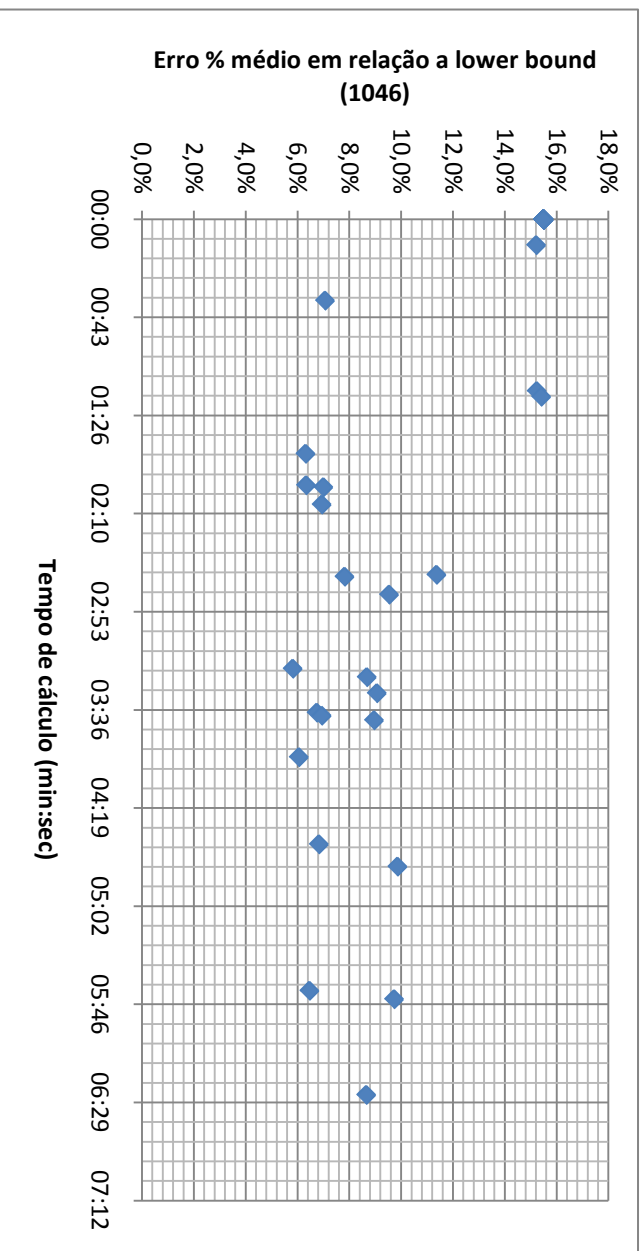


FIGURA E.4 – GRÁFICO RELACIONANDO ERRO % MÉDIO EM RELAÇÃO AO ÓTIMO.

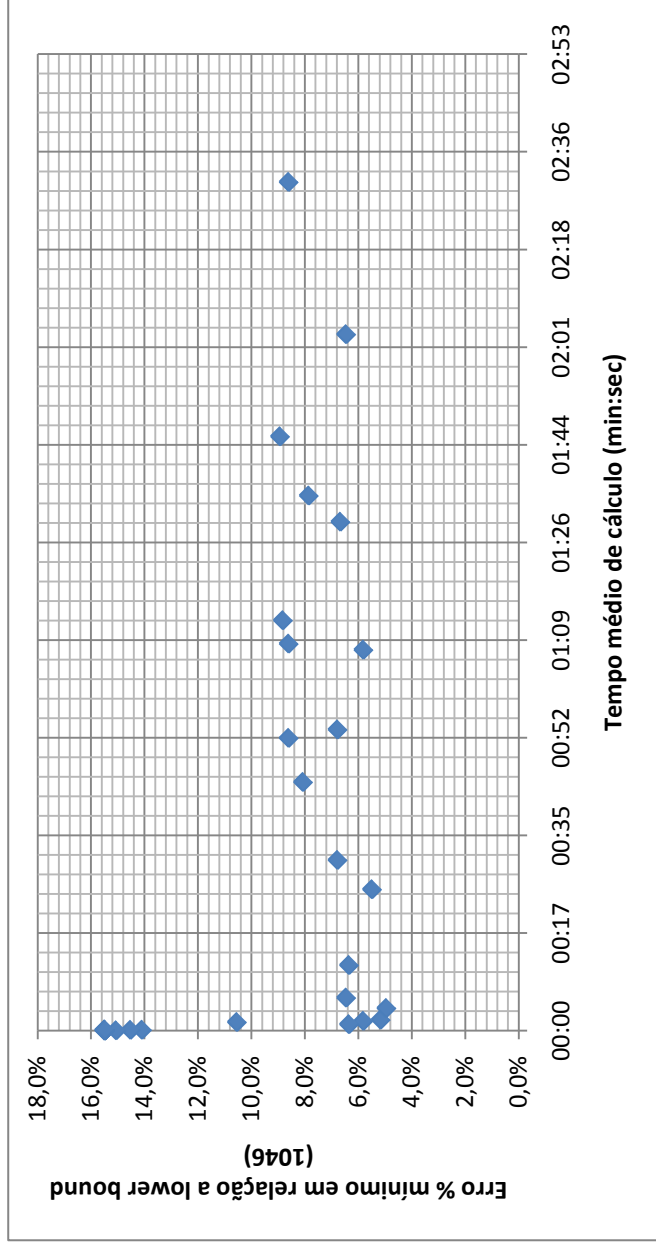


FIGURA E.5 – GRÁFICO RELACIONANDO ERRO % MÍNIMO EM RELAÇÃO AO ÓTIMO.

F. DADOS DA SOLUÇÃO DO PROBLEMA COM ROTAS ALTERNATIVAS – CHAN (CHAN, 2003)



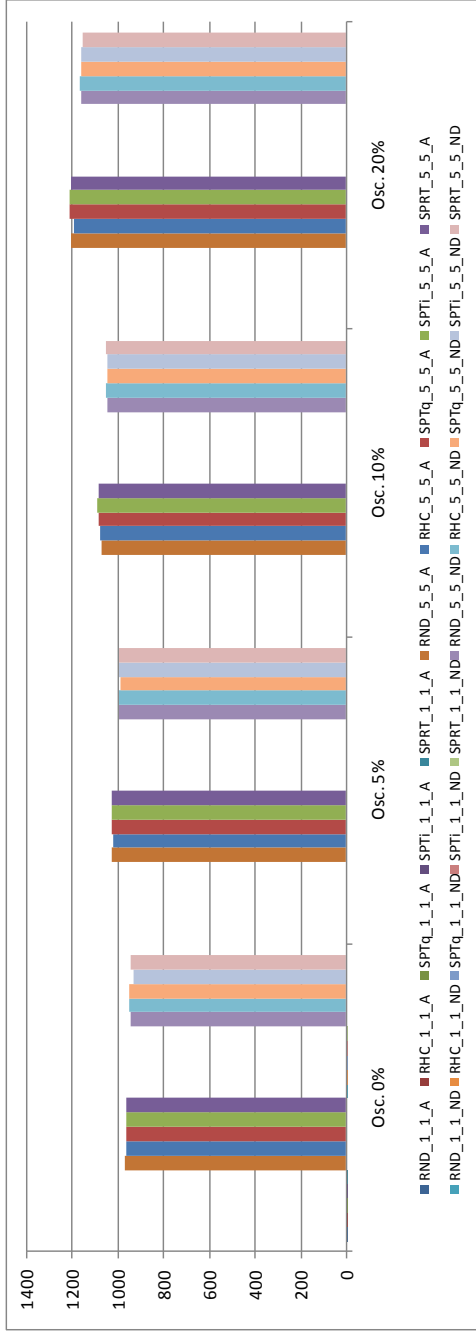


FIGURA F.1 – GRÁFICO RELACIONANDO *MAKESPAN* MÉDIO E OSCILAÇÕES: HEURÍSTICAS DIVERSAS; TAXAS PM DE 1% E 5%; AGENDAMENTOS ATIVO E ATIVO ND.

[illegible]

QUADRO F.2 – RESUMO DA SOLUÇÃO DO PROBLEMA DE CHAN ORIGINAL (CHAN,2003): TEMPOS MÉDIOS E AGENDAMENTOS A E ND COM OSCILAÇÕES DE 0, 5, 10 E 20%.

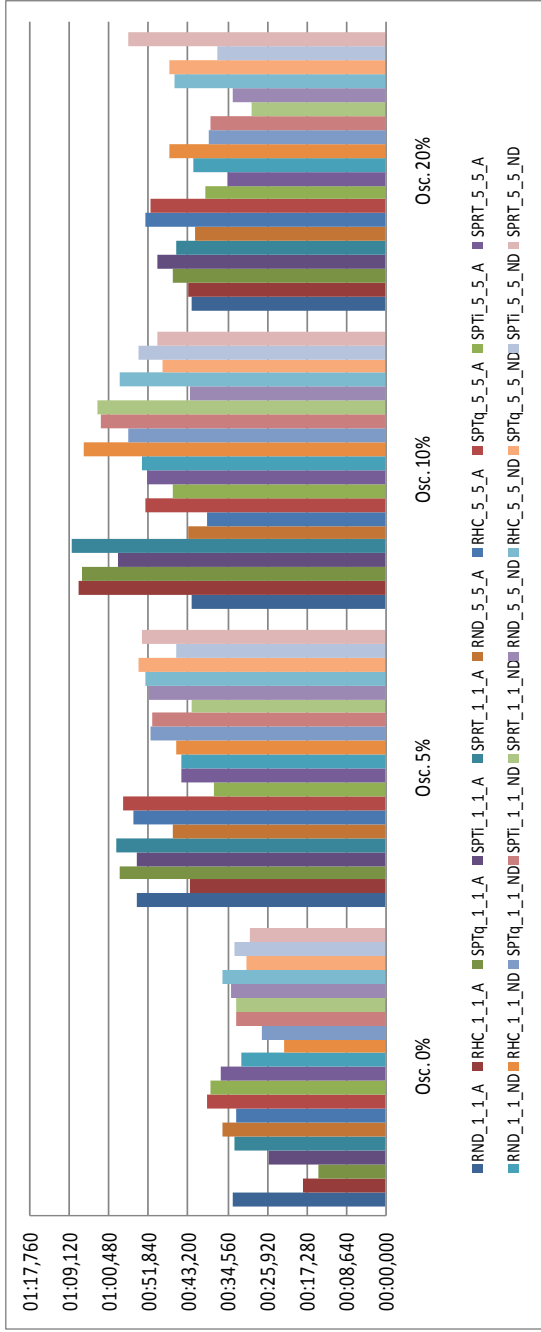


FIGURA F.2- GRÁFICO RELACIONANDO *MAKESPAN* MÉDIO E OSCILAÇÕES: HEURÍSTICAS DIVERSAS; TAXAS PM DE 1% E 5%; AGENDAMENTOS ATIVO E ATIVO ND.



G. DADOS DA SOLUÇÃO DO PROBLEMA COM ROTAS ALTERNATIVAS – CHAN EXPANDIDO (ARAÚJO, 2006)

[illegible]

QUADRO G.1 – RESUMO DA SOLUÇÃO DO PROBLEMA CHAN MODIFICADO (ARAÚJO, 2006): *MAKESPANS* MÉDIOS PARA AGENDAMENTOS A E ND COM OSCILAÇÕES DE 0, 5, 10 E 20%.

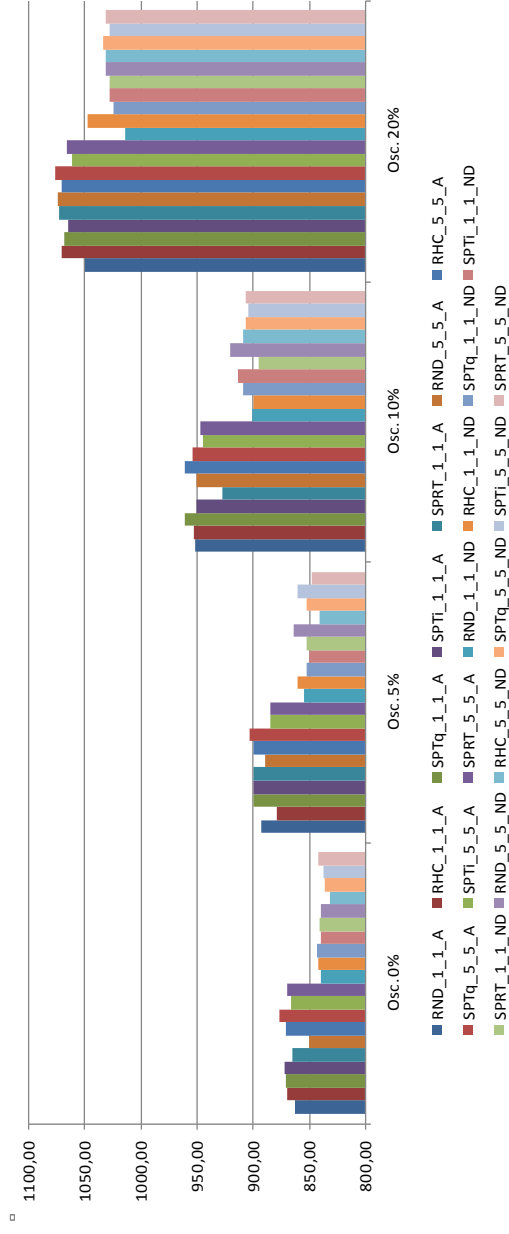


FIGURA G.1 – GRÁFICO RELACIONANDO MAKESPAN MÉDIO E OSCILAÇÕES: HEURÍSTICAS DIVERSAS; TAXAS PM DE 1% E 5%; AGENDAMENTOS ATIVO E ATIVO ND.

Tempo médio (A)													
1% - 1% (A)													
L.B.	Menor méd.	RND	RHC	SPTq	SPTI	SPRT	RND	RHC	SPTq	SPTI	SPRT	SPTI	SPRT
00:00,063	00:20,726	00:32,225	00:24,886	00:20,726	00:27,815	00:33,931	00:24,731	00:24,480	00:26,242	00:28,041	00:29,841	00:29,841	
00:04,483	00:41,315	00:51,799	01:00,923	00:49,147	00:49,477	00:54,538	00:52,185	00:45,519	00:41,315	00:54,395	00:59,127		
00:03,704	00:39,251	00:39,251	00:54,432	00:45,017	00:43,204	00:54,713	00:55,633	00:49,382	00:41,301	00:53,719	00:56,451		
00:00,803	00:34,817	01:01,116	00:50,102	00:52,247	01:12,393	00:46,687	00:41,453	00:49,202	00:34,817	00:44,954	01:02,325		
Tempo médio (ND)													
1% - 1% (ND)													
L.B.	Menor méd.	RND	RHC	SPTq	SPTI	SPRT	RND	RHC	SPTq	SPTI	SPRT	SPTI	SPRT
00:00,298	00:17,310	00:24,912	00:32,641	00:28,393	00:23,818	00:29,094	00:38,345	00:17,310	00:31,117	00:21,087	00:43,337		
00:02,075	00:44,320	00:52,834	00:51,786	00:44,320	01:01,010	00:55,530	00:44,756	01:04,155	00:57,361	00:52,480	00:54,061		
00:05,016	00:39,754	00:49,094	00:59,875	00:54,776	00:51,904	00:48,497	00:43,889	00:39,754	00:52,433	00:49,935	00:50,189		
00:00,828	00:42,654	00:57,363	01:00,636	01:09,173	01:01,087	01:04,451	00:51,049	00:42,654	00:47,865	00:46,110	00:46,768		

QUADRO G.2 – RESUMO DA SOLUÇÃO DO PROBLEMA CHAN MODIFICADO (ARAUJO,2006): TEMPOS MÉDIOS PARA AGENDAMENTOS A E ND COM OSCILAÇÕES DE 0, 5, 10 E 20%.

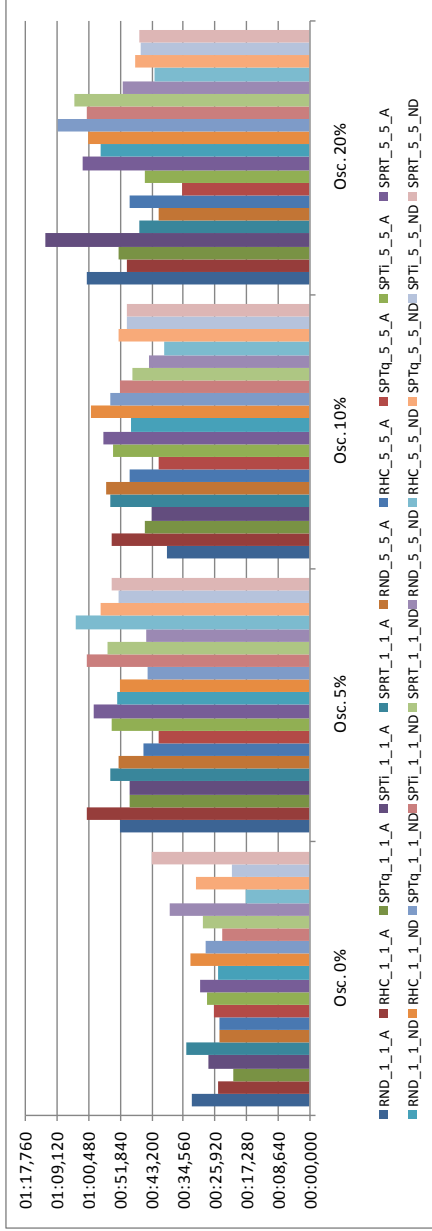


FIGURA G.2 – GRÁFICO RELACIONANDO TEMPOS MÉDIOS E OSCILAÇÕES: HEURÍSTICAS DIVERSAS; TAXAS PM DE 1% E 5%; AGENDAMENTOS ATIVO E ATIVO ND.



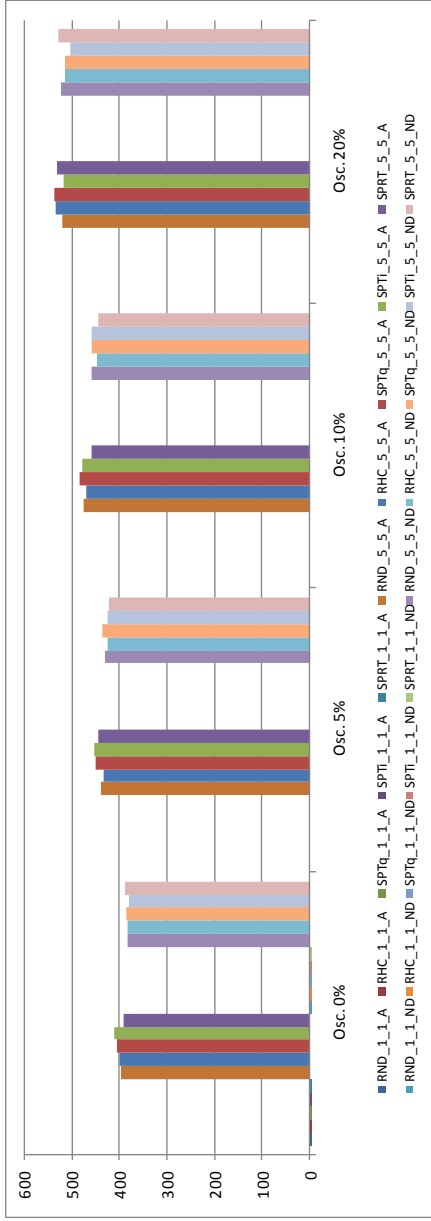


FIGURA H.1 – GRÁFICO RELACIONANDO MAKESPAN MÉDIO E OSCILAÇÕES: HEURÍSTICAS DIVERSAS; TAXAS PM DE 1% E 5%; AGENDAMENTOS ATIVO E ATIVO ND.

[illegible]

QUADRO H.2 – RESUMO DA SOLUÇÃO DO PROBLEMA KUMAR ET AL. (2003): TEMPOS MÉDIOS PARA AGENDAMENTOS A E ND COM OSCILAÇÕES DE 0,5, 10 E 20%.

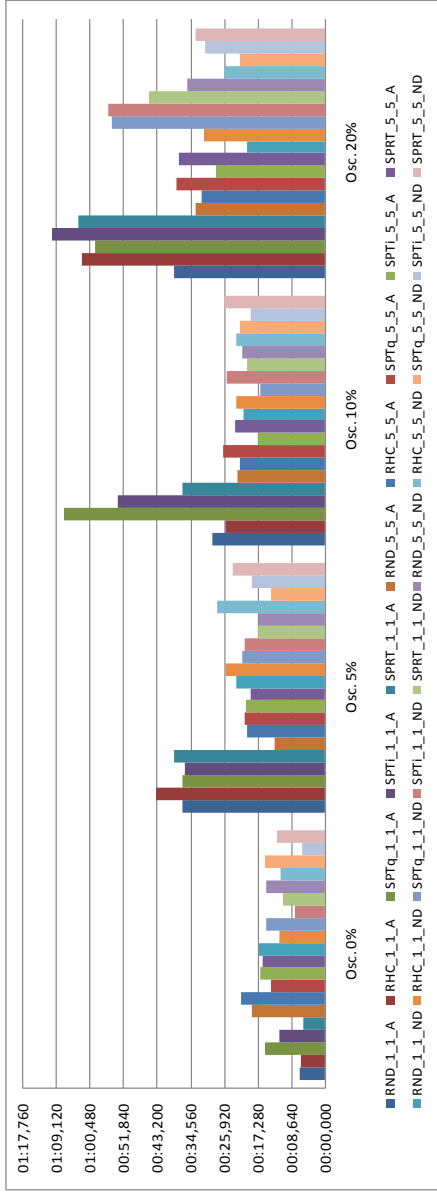


FIGURA H.2 – GRÁFICO RELACIONANDO TEMPO MÉDIO E OSCILAÇÕES, PARA DIVERSAS HEURÍSTICAS, TAXAS PM DE 1% E 5%, ALÉM DOS AGENDAMENTOS ATIVO E ATIVO ND.